D-A095 735	ANALYS DEC 80	NGINEER IS OF A J B P /GL-80-	N ASSEM ALMERTO	AYS EXP	PERIMENT OF DISCS	STATION EMPLOY	YING IN	SBURG-	VE GRAP	/6 9/2 HIETC NL	(U)	
104 1844 13	entra de	/										
											END OATE FILMED	



70

AD A 0957



MISCELLANEOUS PAPER GL-80-9

ANALYSIS OF AN ASSEMBLAGE OF DISCS EMPLOYING INTERACTIVE GRAPHICS

Ьу

John B. Palmerton

Geotechnical Laboratory
U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Miss. 39180

SELECTE MAR 0 2 1981

E

December 1980 Final Report

Approved For Public Release; Distribution Unlimited



Prepared for Assistant Secretary of the Army (R&D)

Department of the Army

Washington, D. C. 20310

Under Project 4A161101A91D Task 02, Work Unit 120

81 3 2 027

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

14[WEZ/W/ 1]

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

	REPORT DOCUMENTATION PAGE		
. REPORT NUMBER		BEFORE COMPLETING FORM NO. 3. RECIPIENT'S CATALOG NUMBER	
Miscellaneous Paper GL-80-9	AD-A09513	<u>'5 </u>	
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED	
ANALYSIS OF AN ASSEMBLAGE OF DI INTERACTIVE GRAPHICS.	SCS EMPLOYING	Final reports I 11-5-6	
· · · · · ·			
7. AUTHOR(a)	·····	8. CONTRACT OR GRANT NUMBER(*)	
John B./Palmerton \		1) Des	
9. PERFORMING ORGANIZATION NAME AND ADD		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
U. S. Army Engineer Waterways E	Experiment Station	Project/4A161101A91D1	
Geotechnical Laboratory	20100	Task 02, Work Unit 120	
P. O. Box 631, Vicksburg, Miss. D. Controlling Office NAME and ADDRESS			
Assistant Secretary of the Army		2. REPORT DATE December 1980	
Department of the Army	7-100 V	13. NUMBER OF PAGES	
Washington, D. C. 20310		89	
14. MONITORING AGENCY NAME & ADDRESS(II di	illerent from Controlling Offic	e) 15. SECURITY CLASS. (of this report)	
		Unclassified	
		SCHEDULE	
17 DISTRIBUTION STATEMENT (of the abetract or	ntered in Block 20, if differen	t from Report)	
18. SUPPLEMENTARY NOTES			
9. KEY WORDS (Continue on reverse side it necess	eary and identify by block num	ber)	
19. KEY WORDS (Continue on reverse elde it necess Computer programs ols(eary and identify by block num	ber)	
Computer programs	eary and identify by block num	ber)	
Computer programs 0180 Computer graphics 0. ABSTRACT (Continue on reverse side H necesse)	ary and identity by block num	9 er)	
Computer programs 0180 Computer graphics 0. ABSTRACT (Continue on reverse side H necesse)	development of a epts of the distinct cles (elements) and educations of a education of a educati	computer program entitled DISC. net element method in that the re faithfully represented. The rollection of individual ele-	

DD 1 JAN 73 1473 EDITION OF THOU 65 IS OBSOLETE

Unclassified SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4/2 1 4.

300

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued).

material properties apply throughout the system. The types of particles that can be accommodated by DISC consist of discs and bar-shaped elements. These particles are permitted to react with one another by touching, rolling, bouncing, or sliding. Problems requiring a quasi-static solution or problems involving large velocities and displacement can be analyzed.

In addition to extending the distinct element method to disc-shaped particles, extensive use was made of interactive graphics. Through interactive graphics, the user of the program is relieved of the usual difficulties involved in data preparation and the plotting of results. The mode of operation of the computer program is visual; i.e., pictures (or plots) of the system of disc-shaped elements are drawn on a cathode ray terminal. The program operates in a time-sharing environment.

Acces	sion For	•		
NTIS	GRA&I	X		
DTIC TAB				
Unannounced 🔲				
Justification				
By				
Distribution/				
Availability Codes				
	Avail a	n d/or		
Dist	Specia	a l		
A		ا		

PREFACE

This study was authorized as part of the In-House Laboratory Independent Research (ILIR) Program for FY 78 and FY 79 and was performed under Project 4A161101A91D, Task 02, Work Unit 120, sponsored by the Assistant Secretary of the Army (R&D). The investigation was conducted during the period December 1977 to September 1980 at the U. S. Army Engineer Waterways Experiment Station (WES).

The study was conceived and conducted by Mr. J. B. Palmerton, Research Civil Engineer, Engineering Geology and Rock Mechanics Division (EGRMD), Geotechnical Laboratory (GL), under the general supervision of Dr. D. C. Banks, Chief, EGRMD, Mr. J. P. Sale, former Chief, GL, and Dr. P. F. Hadala, Assistant Chief, GL. This report was prepared by Mr. Palmerton.

Commanders and Directors of the WES during the conduct of this study were COL John L. Cannon, CE, and COL Nelson P. Conover, CE. Technical Director was Mr. Fred R. Brown.

CONTENTS

			Page
PREFACE			1
PART I: INTRODUCTION		•	3
Background	•		3
Purpose	•	•	3
PART II: MATHEMATICAL CONCEPTS	•	•	4
PART III: PROGRAM ORGANIZATION	•	•	13
General Overview			13
Main Program		•	15
Subroutine GIDYUP	•	•	16
Subroutine INPUT			16
Subroutine OUTPUT			17
Subroutine PLIST		•	21
Subroutine MOTION	•	•	26
Other Subroutines	•	•	51
PART IV: PROGRAM OPERATION		•	54
PART V: CONCLUSIONS AND RECOMMENDATIONS			75
Conclusions			75
Recommendations	•	•	76
REFERENCES		•	77
APPENDIX A: SCHEMATIC DIAGRAM OF DISC AND FORTRAN			
SOURCE LISTING		_	A 1

ANALYSIS OF AN ASSEMBLAGE OF DISCS EMPLOYING INTERACTIVE GRAPHICS

PART I: INTRODUCTION

Background

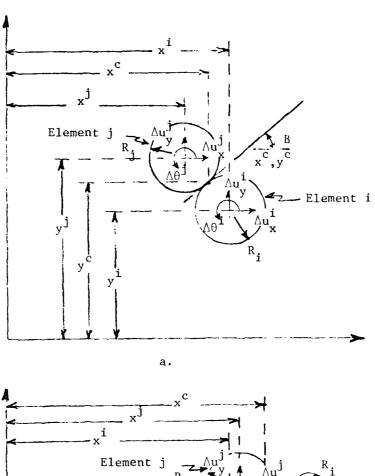
1. A computer modeling technique termed the "distinct" element method was introduced by Cundail (1971). Then, several years later, the technique was extended to analyze blocky rock systems (Cundail, 1974). This type of analysis permitted large-scale motions of the individual rock blocks. This work was further extended to the analysis of tunnel supports (Voegele, 1979). Thus, the distinct element method has been developed to accurately model many features of the behavior of jointed rock.

Purpose

- 2. This study was directed at developing methods for analyzing the behavior of systems of simple, particulate members. The basic shape of the particulate members was chosen to be a disc (or cylinder). The distinct element method is oriented more toward representing the kinematics of a system of particulate members instead of the continuum aspects. At the outset, the structure is viewed as a collection of individual members with individual properties, rather than a homogeneous system for which the material property effects are assumed to apply throughout.
- 3. In addition to extending the distinct element method to discs, there was a desire to use computer interactive graphics in the analysis so that the user could be relieved of laborious data preparation and time-consuming plotting of the results. This report describes the development of a computer program entitled DISC and contains discussions of the mathematical formulations, program organization, and program operation.

PART 11: MATHEMATICAL CONCEPTS

- 4. The computer program DISC was formulated using concepts of the "distinct element" method proposed by Cundall (1974). Conceptually, the distinct element method computes the forces and displacements at points of contact between mathematically described particles (elements). The contact between the elements are mathematically represented by springs and dashpots. Any relative displacement between two elements in contact then results in a force developing within the springs and dashpots. The vector sum of all such forces on the discs in turn causes accelerations of the discs. If these accelerations are assumed to be constant for a prescribed interval of time, the associated velocities and displacements may be computed for the end of that time interval. These new displacements may then be used to determine new relative displacements between the contacting elements, and the cycle is then repeated over and over.
- 5. The computer program DISC is capable of handling element shapes corresponding to discs and bars. The bar elements are composed of two parallel sides of arbitrary length terminated on both ends by semicircles. Figure 1 shows two examples of two elements in contact. Condition a presents two disc elements in contact, and condition b a bar and a disc element in contact. Mathematically, the equations required to describe the subsequent motions of either contact pair are identical. However, condition b is somewhat more general and easier to visualize and as such will be referred to in the following discussion.
- 6. Figure 1b shows the bar element i, its centroid located at x^i , y^i , and the disc element j, its centroid located at x^j , y^j . The coordinates of the contact point are at x^c , y^c . The angle to the contact plane is denoted by B. (For the conditions shown in Figure 1a, B is the angle of the plane of tangency.)
- 7. The basic key equation of the distinct element method relates the incremental displacement of the contact point (into one of the two contacting bodies) to the relative motion of the elements that are in contact. The following equation is the result of superposition, i.e.,



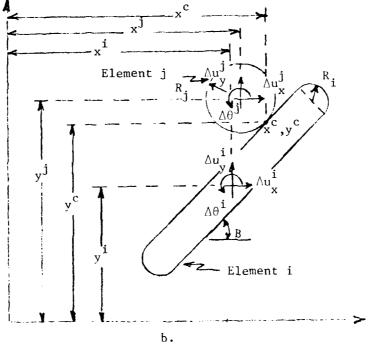


Figure 1. Basic elements and definitions

first assume element i is fixed and determine the resulting motion of the contact, and then assume element j is fixed and examine the motion of the contact point. This analysis yields (as illustrated in Figure 2a)

$$\Delta u_y^c = \Delta u_y^j - \Delta u_y^i + \Delta \theta^j (x^c - x^j) - \Delta \theta^i (x^c - x^i)$$
(1)

and

$$\Delta u_{x}^{c} = \Delta u_{x}^{j} - \Delta u_{x}^{i} - \Delta \theta^{j} (y^{c} - y^{j}) + \Delta \theta^{i} (y^{c} - y^{i})$$

where

 $\Delta u_y^c \; = \; the \; vertical \; component \; of \; the \; incremental \; contact \\ displacement$

 Δu_{x}^{c} = the horizontal component of the incremental contact displacement

 Δu_y^i = the current vertical incremental displacement of element i, etc. (i.e., the subscript (x or y) refers to the component direction and the superscript (i or j) refers to the element)

 x^{i} , y^{i} = the centroid coordinates of element i

 x^{j} , y^{j} = the centroid coordinates of element j

 x^{c} = the horizontal coordinate of the contact point

 y^{c} = the vertical coordinate of the contact point

 $\Lambda \theta$ = the incremental rotation of the element

8. The global angle of the contact plane is given by B. Thus, the horizontal and vertical relative contact displacements may be resolved into components parallel and perpendicular (shear and normal directions) to the contact plane. Thus, as shown in Figure 2b

$$\Delta u_{s}^{c} = \Delta u_{y}^{c} \sin B + \Delta u_{x}^{c} \cos B$$
 (2)

and

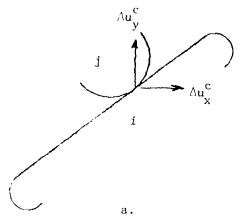
$$\Delta u_n^c = \Delta u_y^c \cos B - \Delta u_x^c \sin B$$

where

 Δu_{S}^{c} = the component of the contact point displacement parallel to the contact plane

 Δu_n^c = the normal component

9. As previously mentioned, the contacts are represented by springs and dashpots (in both the normal and shear directions). Thus,

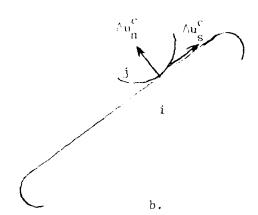


$$\Delta u_y^c = \Delta u_y^j - \Delta u_y^i + \Delta \theta^j (x^c - x^j)$$

$$-\Delta \theta^i (x^c - x^i)$$

$$\Delta u_x^c = \Delta u_x^j - \Delta u_x^i - \Delta \theta^j (y^c - y^j)$$

$$+ \Delta \theta^i (y^c - y^i).$$



$$\Delta u_{s}^{c} = \Delta u_{y}^{c} \sin B + \Delta u_{x}^{c} \cos B$$
$$\Delta u_{n}^{c} = \Delta u_{y}^{c} \cos B - \Delta u_{x}^{c} \sin B$$

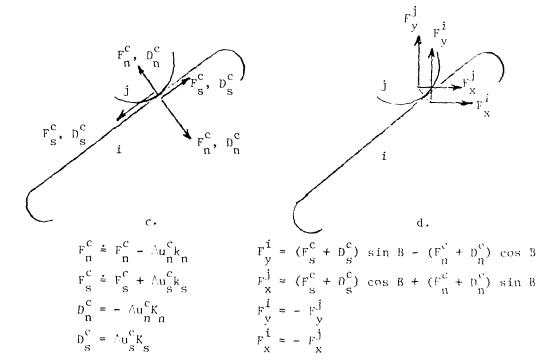


Figure 2. Contact displacements and forces

the force generated within the springs and dashpots because of the incremental normal and shear displacements is given by

$$F_{n}^{c} \doteq F_{n}^{c} - \Delta u_{n}^{c} k_{n}$$

$$F_{s}^{c} \doteq F_{s}^{c} + \Delta u_{s}^{c} k_{s}$$

$$D_{n}^{c} = -\Delta u_{n}^{c} K_{n}$$

$$D_{s}^{c} = \Delta u_{s}^{c} K_{s}$$
(3)

where

 F_n^c = the normal spring force

 F_e^c = the shear spring force

 D_{p}^{c} = the normal dashpot force

 D_c^C = the shear dashpot force

 k_n = the normal stiffness

 k_{g} = the shear stiffness

 $K_n =$ the normal damping constant

 $K_{\rm e}$ = the shear damping constant

The symbol $\stackrel{.}{=}$ means "replaced by"; that is, after each increment of contact point displacement, new normal and shear forces are recomputed as functions of the old values. The dashpots are necessary to prevent the two contact elements from vibrating indefinitely. To be strictly correct, the dashpot forces should be related to the contact point velocity; however, for a time increment, the incremental displacement is proportional to the velocity (i.e., $\Delta u = v\Delta t$, where v is the relative velocity across the contact). Figure 2c shows the positive directions of the normal and shear forces. Also, note that a positive normal force indicates a compressive force.

10. Equation 3 is subject to modification if the tensile "strength" T (a negative value) is exceeded. Thus, it follows that if

$$F_n^c < T$$

then

$$F_n^c = F_s^c = D_n^c = D_s^c = 0$$

If F_n^c is less than T (which is normally set to zero), all forces at the contact are set to zero, i.e., the elements are tending to separate. In addition, if the coefficient of friction is given as μ , the equations are modified as follows:

Ιf

$$\left| F_{s}^{c} \right| > \mu F_{n}^{c}$$

then

$$F_s^c = \mu F_n^c / F_s^c / F_s^c$$

$$D_{S}^{C} = 0$$

Thus, the two elements will slide whenever the shear force exceeds the product of the coefficient of friction and the normal force, and the resulting shear force is maintained at the shear "strength."

11. Now that the contact shear and normal forces have been computed, all that remains is to resolve these forces back into the component directions. The result is

$$F_{y}^{j} = (F_{S}^{c} + D_{S}^{c}) \sin B - (F_{n}^{c} + D_{n}^{c}) \cos B$$

$$F_{x}^{j} = (F_{S}^{c} + D_{S}^{c}) \cos B + (F_{n}^{c} + D_{n}^{c}) \sin B$$

$$F_{y}^{i} = -F_{y}^{j}$$

$$F_{y}^{i} = -F_{x}^{j}$$

$$(4)$$

where F_y^j is the vertical component of the contact force on element j, etc. Figure 2d shows the positive directions of the component forces. Obviously, the forces on element i are numerically equal but opposite to those on element j. Up to this point, only a single pair of contacting elements has been considered. In actuality, any element, say

element i, may possess a number of contacts. In addition, element i may be subject to applied forces and gravity forces. Thus, the total force acting on element i is given by

$$F_{ysum}^{i} = \Sigma_{c} F_{y}^{i} + \text{vertical applied load - gravity load}$$

$$F_{xsum}^{i} = \Sigma_{c} F_{x}^{i} + \text{horizontal applied load}$$

$$M_{sum}^{i} = \Sigma_{c} \left[F_{y}^{i} (x^{c} - x^{i}) - F_{x}^{i} (y^{c} - y^{i}) \right] + \text{applied moment}$$
(5)

The symbol $\frac{\pi}{c}$ means the summation for all contact points of element i. The equations are similar for element j. The term $M^{\hat{i}}_{sum}$ is the total moment acting on element i. Figure 3 illustrates the force sums.

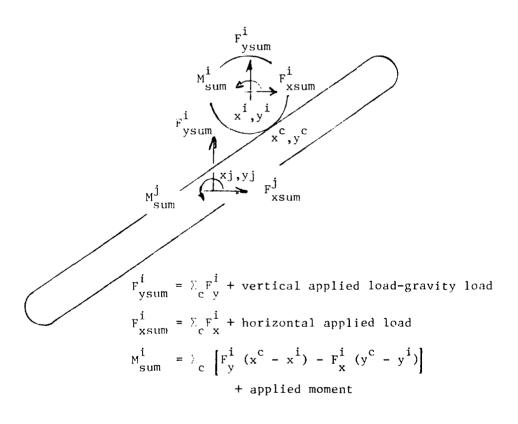


Figure 3. Force sums on element i

12. After computing all the force sums acting on each element, the acceleration of each element may be computed from Newton's law of motion, i.e., acceleration = unbalanced force/mass, and angular acceleration = unbalanced moment/moment of inertia. Thus, the component accelerations a, and angular acceleration α are expressed as

$$a_{x}^{i} = F_{xsum}^{i}/m^{i}$$

$$a_{y}^{i} = F_{ysum}^{i}/m^{i}$$

$$\alpha^{i} = M_{sum}^{i}/I^{i}$$
(6)

where

 m^{i} = the mass of element i I^{i} = the mass moment of inertia of element i

The velocities (ω is the angular velocity) are then computed by multiplying by the time step Δt :

$$v_{x}^{i} \doteq v_{x}^{i} + a_{x}^{i} \wedge t$$

$$v_{y}^{i} \doteq v_{y}^{i} + a_{y}^{i} \wedge t$$

$$\omega^{i} \doteq \omega^{i} + a_{y}^{i} \wedge t$$
(7)

A second numerical integration yields the incremental displacement for the element to be

$$\Delta u_{x}^{i} = v_{x}^{i} \Delta t$$

$$\Delta u_{y}^{i} = v_{y}^{i} \Delta t$$

$$\Delta c^{i} = \omega^{i} \Delta t$$
(8)

The quantities above are sought in order to recycle the calculations. These quantities are introduced back into Equation 1, and the whole process is repeated.

13. Finally, the total displacement of the element centroids are found from

$$u_{x}^{i} \doteq u_{x}^{i} + \Delta u_{x}^{i}$$

$$u_{y}^{i} \doteq u_{y}^{i} + \Delta u_{y}^{i}$$

$$\theta^{i} \doteq \theta^{i} + \Delta \theta^{i}$$
(9)

- 14. At the outset of each problem, the incremental displacements are, of course, zero. Thus, Equations 1 through 4 yield nothing, i.e., there are, up to now, no contact forces. In Equation 5, a force sum will result due to applied and/or gravity forces. Thus, an acceleration of an element will occur that in turn can yield incremental displacements. Equations 1 through 4 are then used to compute the new contact forces.
- 15. As noted from the discussion above, the mathematics involved in program DISC are quite simple, since only the concepts of a damped oscillator and Newton's law of motion are needed. The real difficulty in performing this type of analysis is the development of the logic to detect and to provide computer memory for the many contacts that may result. Thus, the problem becomes one of geometry and bookkeeping. The details of the program are presented in Part III.

PART III: PROGRAM ORGANIZATION

General Overview

- that consists of a main program DISC is a time-sharing FORTRAN program that consists of a main program and sixteen subroutines. Physically, DiSC is composed of approximately 1200 FORTRAN statements. The program is written to use interactive graphics and is operated by a Tektronix 4014 (or 4010) terminal. With the exception of occasionally typing in a few numbers to redefine certain parameters, the commands primarily consist of single keystrokes. The output is primarily graphical, i.e., pictures of the present locations of the elements are drawn on the screen. While it is possible to obtain printed information, the normal mode of operation is visual.
- 17. As discussed in Part II, the actual calculations for determining the motion of the disc and bar elements are quite straightforward. Enowing the location of a contact point and the previous relative displacements of those elements in contact, the forces between the elements can be computed and then integrated over time to obtain new displacements necessary for the next cycle. This process is carried out for all contacts at each time cycle.
- 18. Suppose though, that a system of 100 disc elements, packed closely together, is to be considered. If the discr are of the same size, then there could, at most, be six contacts per disc; and since some must be on or near the edge, the total number of contact points must be less than 600. However, in order to check for contacts between the two arbitrary discs, it would (by brute force) be necessary to check for the possibility of a contact between each and every disc. Furthermore, an inordinate amount of searching would be required. If n = 100 discs, the number of different possible combinations.

$$C = \frac{n!}{2(n-2)!} = \frac{100!}{2(98!)} = 4950 \tag{10}$$

For 500 discs, the number of searches would increase to 124,750. In addition, computer memory space would be required at double the amounts above to store the normal and shear contact forces.

- 19. Since most of this memory would be blank and most of the searches fruitless (where contacts were not found), a great deal of effort was put into formulating schemes and methods to facilitate the program's efficiency. Indeed, the usability of the distinct element method is predicated on efficient programming techniques. Without incorporating these schemes it would not be possible to solve even small problems on large computers.
- 20. The various subroutines that make up the program DISC are related to each other primarily through named common storage. The items in common storage are as follows:

N = Number of disc and bar elements

SKN = Spring stiffness at contact

DKN = Damping constant of contacts

TIME = Time since beginning of run

DT = Time increment between steps

DF = Damping factor (a multiplier for the damping constant)

FXSUM(M) = Sum of forces in x-direction on element M

FYSUM(M) = Sum of forces in y-direction on element M

FMSUM(M) = Sum of moments acting on element M

FX(M) = Applied x-force on element M

FY(M) = Applied y-force on element M

FM(M) = Applied moment on element M

X(M) = Current x-coordinate of element M

Y(M) = Current y-coordinate of element M

T(M) = Current angelar rotation of element M

R(M) = Radius (or thickness, if bar element) of element M

S(M) = One half of the separation between the circular ends of bar element M

W(M) = Weight of element M

XM(M) = Mass of element M

UX(M) = Current x-displacement of element M

UY(M) = Current y-displacement of element M

UT(M) = Current angular displacement of element M

VX(M) = Current x-velocity of element M

VY(M) = Current y-velocity of element M

 $\mathsf{IFIX}(\mathsf{M})$ = Fixity code for element M . The values are:

1 = Fixed in x-direction

2 = Fixed in y-direction

3 = Fixed in x- and y-direction

4 = Fixed in angular direction

5 =Fixed in x and angular directions

6 = Fixed in y and angular directions

7 = Fixed in all directions

8 = A constant prescribed velocity applies

TEN(M) = Tensile strength at contact for element M

 $AA(M \times 24)$ = AA is the contact list. See discussion of subroutine MOFION

NA(M) = Index used to find sought quantities in the contact list

LL (M \times 6) = Possible contact search list. See discussion of subroutine PLIST

The common scorage requires approximately 53 N storage locations.

Main Program

21. The main program initiates the graphics software and calls subroutine GEDYUP. Control is never returned to the main program. The graphics subroutines (West Point Military Academy, 1975) are peculiar to the WES computing system and also the Office of Personnel Management's Honeywell system at Macon, Georgia. The graphics subroutines employed in DISC include:

USTART	UBELL	UCRCLE
USET	UERASE	UPEN
UWATT	UPRINT	UMOVE
UREAD	UPRNT1	UPSET
UGRIN	UARC	UDAREA
UWINDO	UROTATE	UOUTLN

Subroutine GIDYUP

- 22. Subroutine GIDYUP is the driving subroutine. Its primary function is to call subroutine INPUT and to cause cycling through subroutine OUTPUT. This subroutine is the only one that calls the main calculation subroutine MOTION. The normal sequence of operations involves:
 - a. GIDYUP calls INPUT. The user interactively describes the problem. Alternatively, the user may select a problem previously stored by subroutine SAVE.
 - b. A call is made to PLIST, which prepares a list of those elements for which a contact is possible within the next 50 cycles.
 - c. A call is made to MOTION, which calculates the movements of and forces on the elements. Fifty cycles are performed in MOTION. At the end of each 50 cycles, the new positions of the elements are graphically plotted by OUTPUT. Step b is again performed.
 - d. Every so often (user selected, default value = 250 cycles) a call is made to OUTPUT. The program operation is halted, and the user must inform the system what action is to be taken next (i.e., continue running, draw all elements, change parameters, etc.).

Subroutine INPUT

23. Subroutine INPUT is used to describe the problem to be analyzed (see user's guides (Figures 9 and 10)) and also to initialize common data storage dependent upon its calling subroutine. It may be called by GIDYUP (to start a new problem) or by OUTPUT (to modify an existing problem). The following subroutines may be called by INPUT:

SCREEN CIRCLE
GIDYUP LOOP
GRID GEN
SAVE

Subroutine OUTPUT

- 24. A variety of functions can be performed by subroutine OUTPUT (see user's guides (Figures 9 and 10)). After the problem geometry is described in INPUT, subroutine OUTPUT takes over the system operation. Its function is to set boundary conditions (fix certain elements), to allow for inputting parameters (weights, damping factors, frictional properties, etc.), and to handle both graphical and formatted output. Between the phases when calculations of motion are being performed between GIDYUP and MOTION, the program halts occasionally (user defined, default value = 250 cycles) in OUTPUT. During this halt, the user may redefine boundary conditions, parameters, etc., plot the data, or request another calculation phase.
- 25. During any program halt, the user may also instruct the program to store a copy of the existing data geometry (performed by subroutine SAVE). This stored data may then be recalled by a direct call to INPUT. Subroutine OUTPUT is also used to delete (crase) certain elements; however, elements may not be added in OUTPUT. At the outset of a problem, a great deal of geometrical editing may be accomplished by juggling the program back and forth between OUTPUT and INPUT. Consequently, the user is able to reposition elements, add elements, delete elements, change boundary conditions, etc.
- 26. The data storage (SAVE) feature is very handy for situations in which it is desired to solve a suite of problems with the same geometry but different parameters. After one problem is solved, the stored data may be recalled, the parameters changed, and the next problem solved.
- 27. Subroutine OUTPUT may be called only by subroutine GIDYUP. The following subroutines may be called by OUTPUT.

GIDYUP	CIRCLE	DAMP
INPUi	VECTOR	PLTFOR
LOOP	WEIGHT	GRID
SCREEN	INTERVAL	SAVE

- 28. Subroutine OUTPUT is also used to compute the contact stiffness k and time-step increment Δt . (Subroutine INPUT tentatively sets k and Δt in the same fashion as OUTPUT. However, OUTPUT redefines a stable time step dependent upon the boundary conditions.)

$$m \frac{d^2 u}{dt^2} + ku = 0$$

where u is the displacement. In a finice difference notation, the derivative d^2u/dt^2 may be expressed as

$$\frac{d^{2}u}{dt} = \frac{u_{t+1} - 2u_{t} + u_{t-1}}{At^{2}}$$

where \mathbf{u}_{t+1} , \mathbf{u}_{t} , and \mathbf{u}_{t-1} represent the displacements at times t+1, t, and t-1, respectively. Substitution into the equation of motion results in

$$u_{t+1} + \left(\frac{k}{m} \Delta t^2 - 2\right) u_t + u_{t-1} = 0$$

The equality of the finite difference equation above can hold only if the quantity $\left(\frac{k}{m} \Delta t^2 - 2\right)$ is negative. Thus, the stable time step Δt is given by

$$\Delta t \leq \sqrt{\frac{2m}{k}} \tag{11}$$

Greater values for $\mbox{$\Lambdat}$ will lead to exponentially increasing $\mbox{$u$}$.

30. It can be shown that the formulation for DISC (when considering a single normal contact) yields the same finite difference equation given above. That is, given (for some time step t) the values of \mathbf{u}_t , $\Delta\mathbf{u}_t$, \mathbf{v}_t , and \mathbf{f}_t in the normal directions, Equation 3 becomes

$$F_{t+1} = F_t - k\Delta u_t = -\sum_{t=0}^{t} k\Delta u_t = -ku_t$$

Equations 6 and 7 may be expressed as

$$v_{t+1} = v_t + \frac{F_{t+1}^{\Delta t}}{m} - v_t - \frac{ku\Delta t}{m}$$

From Equation 8 and by substitution of the preceding equation

$$\Delta u_{t+1} = v_{t+1} \Delta t = v_t \Delta t - \frac{k u_t \Delta t^2}{m}$$

Then, making the substitution

$$v_t \Delta t = \Delta u_t = u_t - u_{t-1}$$

results in

$$\Delta u_{t+1} = u_t - u_{t-1} - \frac{ku_t \Delta t^2}{m}$$

Finally, Equation 9 may be written as

$$u_{t+1} = u_{t} + (u_{t+1})$$

$$= u_{t} + (u_{t} - u_{t+1} - \frac{ku_{t}/t^{2}}{m})$$

$$= -(\frac{k}{m})t^{2} - 2(u_{t} - u_{t+1})$$

Or by rearranging terms

$$u_{t+1} + \left(\frac{k}{m} At^2 - 2\right) u_t - u_{t-1} = 0$$

The displacement at time t+1 can be computed if the two previous displacements at times t and t+1 are known.

31. The procedure outlined for the computations of a stable time step. At considered only one contact. In actuality, many contacts can be made for a single element; thus, the useful time step must be chosen smaller than the single contact time step. That is, most problems consist of many degrees of freedom.

- 32. Thus, a stable Δt will increase as the mass increases and will decrease as the stiffness increases. The program will, of course, calculate faster (i.e., the cost will go down) as the time step increases. Alternatively, if a Δt is chosen, then a stable k could be chosen for a given m. In general, any two of Δt , k, and m can be chosen, and the other computed to ensure stability through the preceding equation. The scheme incorporated into program DISC for computing a stable Δt is described in the following paragraphs. Any subsequent user of DISC should not necessarily feel bound to this scheme.
- 33. At the outset of writing DISC, it was decided that a disc element with a radius of 20 screen units would be the "standard" disc element. The Tektronix screen is 1023 screen units wide and 780 screen units high. For this standard element, a weight of 100.0 units is assigned. To determine the weight of other disc and bar elements, the area of the element is computed, divided by the standard area, and multiplied by 100.0. Thus, all forces output by the program are in terms of the 100.0 units assigned to the standard disc, i.e., the forces are normalized and, as such, are nondimensional. The element mass me is determined from

$$m_{e} = w_{e}/g \tag{12}$$

where

 w_{α} = the normalized weight of the element

g = the acceleration due to gravity

The value g is an input parameter (default value = 32.2). Thus, if the default value of 32.2 is used, it implies an acceleration due to gravity of 32.2 screen units/unit of time/unit of time. If time is reckoned in seconds, this would normally suggest that one screen unit equals one foot (i.e., g = 32.2 ft/sec²). Inputting g as 9.8 (with time reckoned in seconds) suggests that one screen unit equals one metre (i.e., g = 9.8 m/sec²). It is through this input parameter g, that a scale can be assigned to the problem.

34. The contact spring stiffness k (currently set equal in normal and shear directions) can be assigned in a variety of ways. If the stiffness were physically known, it could be directly assigned. In most quasi-static problems, however, the purpose of the spring stiffness is to prevent the elements from penetrating too far into each other. This requirement implies choosing a very large k; however, choosing a large k results in a small at . For program DISC, the following equation is used for selecting k:

$$\kappa = \psi_{\text{max}} \sqrt{N+1} \tag{13}$$

where

N =the number of elements is the system

w max = the weight of the largest element (actually the weight
 of the largest element not restrained from all movement
 or "fixed")

35. The time step of is then computed as

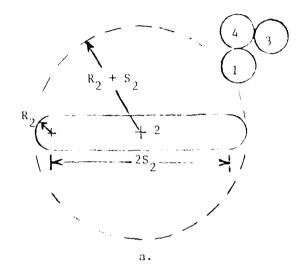
$$\Delta t = 0.05 \sqrt{m_{\min}/k}$$
 (14)

where m_{\min} is the mass of the smallest element not restrained from all movement. The factor 0.05 is used to ensure stability since more than one contact per element is possible. In addition, subroutine OUTPUT computes element areas and moments of inertia and initializes many data lists.

Subroutine PLIST

36. Subroutine PLIST is used to prepare a data list of possible contacts (between elements). This subroutine is accessed before each call to MOTION (i.e., every 50 cycles). Suppose the situation as shown in Figure 4 exists. To determine which elements would go into the possible contact list, a check is first made on the distance dip between each element, i.e.,

$$d_{i,j} = \sqrt{(x_i - x_j)^{\frac{1}{2}} + (y_i - y_j)^{\frac{1}{2}}}$$
 (15)



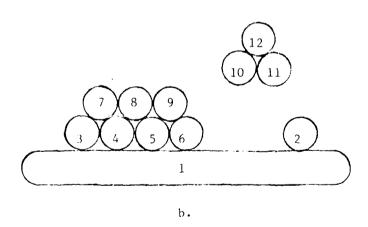


Figure 4. Formation of possible contact list

where $\mathbf{x_i}$, $\mathbf{x_j}$, $\mathbf{y_i}$, $\mathbf{y_j}$ are the coordinates of the element centroids. This distance is compared to the combined element thickness RNG $_{ij}$. The combined element thickness is the sam of the radii (plus lengths, if bar elements) of elements i and j, i.e.,

$$RNG_{i,j} = R_i + R_j + S_i + S_j + 1 \text{ screen unit}$$

$$for a disc element S_i = 0.$$
(16)

One screen unit is added to $RNG_{i,j}$ so that elements that are quite close to contacting will be included in the possible contact list.

37. If $d_{ij} = RNG_{ij}$, then this contact pair is put in the possible contact list. If $d_{ij} = RNG_{ij} + 100$ screen units, the possibility of contact is rejected. If $RNG_{ij} = d_{ij} = (RNG_{ij} + 100)$, one further test is required to accept or reject the contact pair. Since a call to PLIST is made only every 50 cycles, it is possible that moving elements could make contact admin the cycling interval. The velocity of each element (io. the fair computation spele) is known at each call to PLIST. This velocity is a made in example an extrapolated position for the element at (i) and the contact by different extrapolated positions of elements. It and the contact by different extrapolated positions of elements it and the decimal velocities v_x^i , v_y^i and v_x^j , v_y^j are given by

$$x_{i}^{e} = x_{i} + 500 \text{ tv}_{x}^{i}$$

$$y_{i}^{e} = y_{i} + 500 \text{ tv}_{y}^{i}$$

$$x_{j}^{e} = x_{j} + 500 \text{ tv}_{x}^{j}$$

$$y_{j}^{e} = y_{j} + 500 \text{ tv}_{y}^{j}$$
(17)

where x_i^e , y_i^e , etc., are the extrapolated element centroid coordinates after 50 time steps. By substituting

$$V_{x} = 50\Delta t (v_{x}^{i} - v_{x}^{j})$$

$$V_{j} = 50\Delta t (v_{y}^{i} - v_{y}^{j})$$
(18)

the new, extrapolated distance d_{ij}^{e} between element centroids becomes

$$d_{ij}^{e} = \sqrt{(x_{i}^{e} - x_{j}^{e})^{2} + (y_{i}^{e} - y_{j}^{e})^{2}}$$

$$= \sqrt{(x_{i}^{e} - x_{j}^{e} + v_{x}^{e})^{2} + (y_{i}^{e} - y_{j}^{e} + v_{j}^{e})^{2}}$$
(19)

Now, if $d_{ij} = d_{ij}^e$ (i.e., the original distance apart is less than the extrapolated distance apart), then the element pair is moving apart and the contact is not possible. However, if $d_{ij} > d_{ij}^e$, the element pair is closing and this pair is entered into the possible contact list provided that $d_{ij}^e \leq RNG_{ij}^e$.

38. At first sight, it would appear that the last test is all that is necessary to accept or reject the contact candidate. In fact, it is; however, to speed up computations, it is more efficient to check for possible contacts in the order described. As soon as a contact candidate is accepted or rejected, additional computations (involving the velocities) are no longer necessary. The idea is to accept or reject with as few computations as possible. It must be mentioned that PLIST forms only a possible contact list. This list is used in subroutine MOTION, where a closer examination is made to determine if contact is actually made. For example, as shown in Figure 4a, the bar element 2 obviously does not make contact with disc element 1. However, it will be included in the possible contact list. Also, elements 1 and 3 may, if they are moving together rapidly enough, be put in the possible contact list, but they may or may not physically come into contact during the next 50 cycles.

39. Searching for possible contacts is done in the following manner. Starting with element i=1 and continuing incrementally to i=n (the total number of elements), distance checks are made with all elements j with j ranging from i+1 to n. Whenever a possible

contact with element i is found, the value j (the number of the element contacting i) is imbedded in the contact list. After all possible contacts with element i have been found, the value i is imbedded in the list to signal the end of all contacts with element i. For example, if it is assumed that elements 1 and 3, shown in Figure 4a, cannot make contact during the next 50 cycles, the possible contact list will be

$$2 - 4 - 1 - 2 - 4 - 3 - 4$$

Thus, element 1 makes possible contacts with 2 and 4, element 2 makes no new contacts (the 1-2 contact has already been found), element 3 makes contact with 4, and of course, element 4 could not possibly make any as yet undetected contacts. The underlined numbers, 1, 2, 3, and 4, are the ascending values of i and signify the end of each segment of the list. Now, if it is assumed that elements 1 and 3 could make contact, the list would be

$$2 - 3 - 4 - 1 - 2 - 4 - 3 - 4$$

For illustrative purposes, consider the arrangement of elements shown in Figure 4b. The possible contact list for the system would be

$$2 - 3 - 4 - 5 - 6 - 1 - 2 - 4 - 7 - 3 - 5 - 7 - 8 - 4 - 6 - 8 - 9 - 5 - 9 - 6 - 8 - 7 - 9 - 8 - 9 - 11 - 12 - 10 - 12 - 11 - 12$$

or a total of 19 contacts. Notice that the numbering is always forward. There is no reason to search backwards as those contacts will already have been found. The list segments are, in general, longer for the lower numbered elements. Within subroutine PLIST, the possible contact list is entitled LL (MLIST). The reserved length of LL is six times the number of elements, which should be adequate for most problems since this length will handle an average of six contacts per element. There is the possibility that the memory reserved could be exceeded if a large number of bar elements are present in the problem to be analyzed. Subroutine PLIST is called only by GIDYUP. No other subroutines are called by PLIST.

Subroutine MOTION

- 40. Subroutine MOTION performs three basic functions: (a) it detects element contacts, (b) it computes the forces acting on each element, and (c) it computes the subsequent displacement of the element due to those forces. Each entry to MOTION results in 50 time step iterations. Almost all of the processor time used during the operation of DISC is used in this subroutine.
- 41. Figure 5 is the flowchart for subroutine MOTION. As previously mentioned in the discussion of subroutine PLIST, a data list of possible contacts is created by PLIST before each call to MOTION. This list contains the elements that are presently touching or could be close enough to touch during the next 50 time cycles (i.e., the duration for each call to MOTION). While within MOTION, only those contact pairs contained in the possible contact list are considered as potential real contacts. It is not difficult to visualize that this list will be quite small compared to a list unrestricted by distance considerations. Indeed it is likely (at least for disc elements) that the possible contact list is a close approximation to the actual contacts.
- 42. Figure 6 shows the various conditions for contacts between disc and bar elements. When describing element contacts, the convention is to state that element i contacts element j, where j is always greater than i. Searches for contacts are always forward, i.e., the lower numbered element is said to contact the higher numbered element, and never vice versa.
- 43. In general, four situations must be considered to detect real element contacts from the <u>possible</u> contact candidates i and j provided by PLIST.
 - <u>a.</u> Both elements i and j are disc elements (Figure 6a); only one contact can result.
 - b. Element i is a bar element and j is a disc element (Figure 6b); only one contact can result.
 - c. Element j is a bar element and i is a disc element (Figure 6c); only one contact can result.

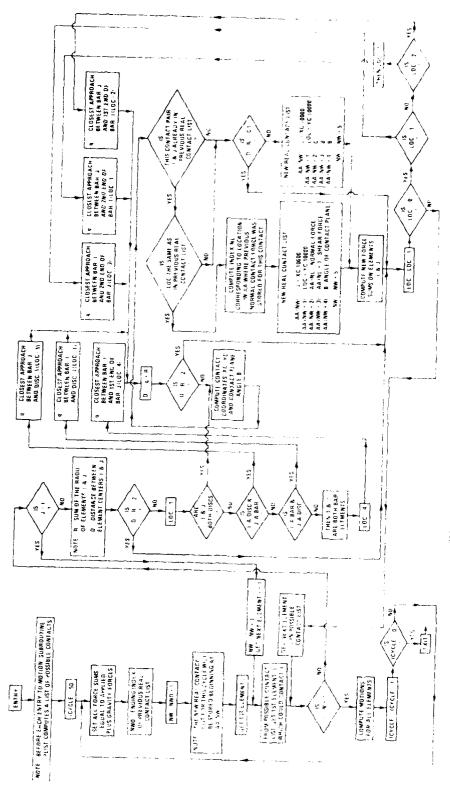


Figure 5. Flow chart of subroutine Molloy

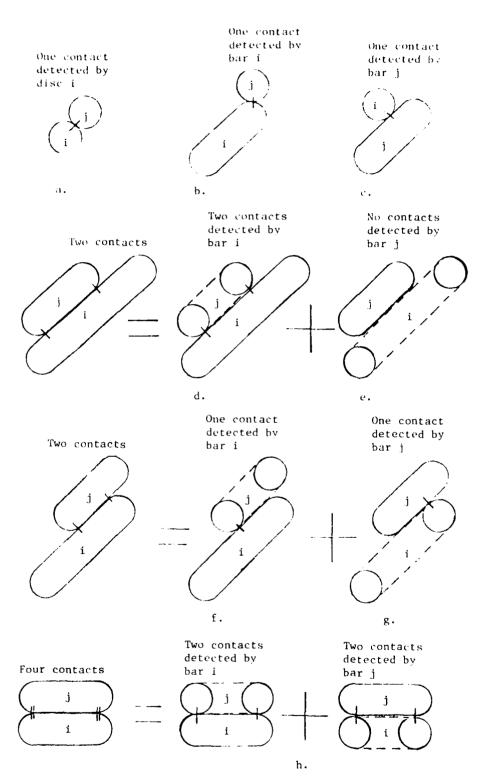


Figure 6. Detection of contacts

- d. Both elements i and j are bar elements (Figures 6d, e, f, g, and h); one, two, or four contacts can result.
- 44. For situation <u>a</u>, it is simply necessary to determine if the disc elements are within the <u>proximity criteria</u> (to be discussed subsequently) in order to be real contacts.
- 45. For situation \underline{b} , determine if disc element j is within the proximity criteria of the periphery of bar element i. If so, a <u>real</u> contact is detected.
- 46. For situation \underline{c} , determine if disc element i is within the proximity criteria of the periphery of bar element j. If so, a real contact is detected.
- 47. For situation \underline{d} , it is necessary to restrict the location of the points of contacts whenever two bar elements are parallel. For two parallel elements, the contact points are defined to be at the locations depicted in Figures 6d, e, f, g, and h, i.e., where circular ends of the elements become tangent to the central, straight portions. All of the contacts may be detected by considering two conditions: (1) element i to be a bar element and element j to be composed of two disc elements (the discs being located at either end of bar j), and (2) element j to be the bar element and element i to be composed of two disc elements.
- 48. Thus, as noted in Figure 6d, the necessary two contacts will be detected if i is considered to be the bar (condition (1) above) and j to be the two discs since both circular ends of bar j touch bar i. However, no contacts would be detected if j is the bar (Figure 6e, condition (2) above) since neither end of bar i makes contact with bar j. Figures 6f and 6g indicate that one contact is detected by choosing i as the bar and the other found by choosing j as the bar. Figure 6h represents the special case of two bars of equal length parallel to one another. In this case, four contacts (two coincident pairs) are detected; two from choosing i as the bar and two from choosing j as the bar.
- 49. For the situation in which two bar elements are not parallel, then only one contact between them is possible. However, the scheme just outlined for detecting contacts between bar elements is also used when examining skewed bar elements. The proximity criteria will reject any potential contacts not sufficiently close.

50. It must be mentioned that no matter whether element i or j or both are considered as the bar element (and the other as composed of two discs) when seeking contacts, the particulars about that contact are always associated with the lower numbered element, i.e., element i. This procedure is in keeping with the "going forward" rule.

Proximity criteria

- 51. The criteria for accepting or rejecting a possible contact are as follows:
 - a. If the peripheries of the two elements are within two screen units of each other at the subject contact point (recall that the "standard" disc element radius is 20 screen units), the contact is accepted as a candidate contact. Otherwise, it is rejected for this iteration cycle.
 - b. After a possible contact is elevated to candidate contact status, two more tests are necessary to finally accept the contact as a real contact. If this candidate contact is present in the contact list generated at the last time step cycle, it is accepted as a real contact and put in the new real contact list. If the candidate contact is not present in the previous contact list, then the two elements must be within 0.1 screen units of each other in order to be placed in the new real contact list. Otherwise, it is rejected for this iteration cycle.

Contact list

52. The contact list is a vector array in COMMON computer storage. This list contains the information about each contact that is necessary for further processing through subroutine MOTION. The contact array is defined by the named COMMON storage statement:

COMMON/STORE/AA(M \times 24), NA(M)

Since M is the maximum number of elements (N is the actual number of elements) that may be considered for a problem, the size of AA is reserved to be at 24 times M. This size for AA should be adequate for most problems since only actual contacts are stored at any given step in the calculations. The contents of AA are constantly changing as the calculations proceed. The array AA should be thought of as divided into two segments: the first segment containing the contact information for

the just completed iteration (the previous contact list), and the second segment containing the information for the current contact list (or the new contact list). As contacts are detected, the contact information is embedded consecutively within the contact list. Since a given element can contact numerous other elements at any given time, the consecutive nature of the list is important to conserve memory. The array NA(i) given above is used to point to the location within AA where the contacts for a given element—i—begin.

53. Before describing the actual structure of the contact list, a discussion of the particular quantities going into the contact list is beneficial. For example, suppose a contact list is being built and the next available location within the array AA is AA(K), where K is the index within common storage. Further, suppose that a contact has just been detected between elements i and j and accepted as a real contact. Also, suppose that this same contact was present in the previous contact list starting at some location AA(L). Thus, the following information is put into the new contact list starting at AA(K):

$$AA(K) = j + x_c/10000$$

 $AA(K + 1) = 10 + y_c/10000$

 $\Delta A(K + 2) = AA(L + 2)$, the previous normal contact force

AA(K + 3) = AA(L + 3), the previous shear contact force

AA(K + 4) = B, the angle from the horizontal to the contact plane

The symbols \mathbf{x}_c and \mathbf{y}_c are the coordinates of the point of contact, and ID is an identifier indicating the fashion in which the contact was detected (i.e., for all cases except two bar elements in contact, ID = 1; for two bar elements, ID takes on values of 1 to 4 depending in which sequence the contacts were found during the search). In effect, locations AA(K) and AA(K + 1) each contain two pieces of information. That is, AA(K) contains as its whole part the element number \mathbf{j} (contacting element \mathbf{j}) and as its fractional part \mathbf{x}_c . A similar situation holds for AA(K + 1).

54. Strictly speaking, only the locations AA(K + 2) and AA(K + 3) contain information (the normal and shear forces on the contact) that

must be retained between iterations in order to accomplish the calculations for displacement. The need for storing the variables $\,j\,$ and $\,$ ID is to aid in determining if the contact now being considered is present in the previous contact list. The variables $\,x_c^{}$, $\,y_c^{}$, and $\,$ B are retained in the contact list as an aid for later outputting of the results.

55. If it is now supposed that the contact being processed was not found present in the previous contact list, the new entry in the contact list would be the same as before, except

$$AA(K + 2) = 0.0$$

$$AA(K + 3) = 0.0$$

That is, the normal and shear contact forces would be set to zero.

discussed was the first contact detected for element i, the contact list pointer NAA(i) is set equal to K. This pointer indicates the position in memory where the information about contacts for element i begins. Now that the contact data for this contact have been entered (and the entry required five memory locations), the next available location to store contact list data is determined by replacing K by K+5 (i.e., $K \doteq K+5$). Any additional detected contacts for element i are, in this manner, put consecutively into the contact list; each detected contact requiring an additional five memory locations. The index K is incremented by five following each entry. After all contacts for element i have been detected and entered into the contact list (and also in the case of no contacts at all being detected for element i), the end of this contact list segment (for element i) is signified by embedding a zero into array AA at location K+1, i.e.,

$$AA(K + 1) = 0.0$$

The embedded zero indicates that this is the end of that segment of the contact list pertaining to element i. The next available location in array AA for storing contact data for the next element $i \doteq i+1$ is now located at $K \doteq K+2$. Since the next element i is now ready to be processed, the pointer NA(i) is set equal to K. The just described process is repeated for all elements.

Initialization of contact data list

57. At the outset of each problem, it is assumed that there are no contacts between the various disc or bar elements. As mentioned in the preceding paragraph, zeros that are embedded in the contact list AA, at the location of the value of the element pointer NA(i), indicate no contacts for element i. Therefore, at the beginning of a problem containing N disc and bar elements, the value of the pointer NA(i) is set to i yielding

NA(1) = 1 NA(2) = 2.
.
.
.
.
.
.
.
.
.
.

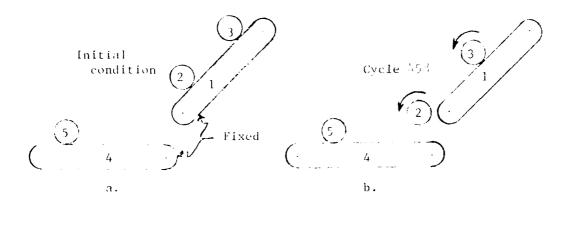
Since a zero located at NA(i) within the array AA indicates no contacts, the array AA(K) is set to zero (for values of K=1, N). In general, the first time cycle in MOTION will result in some contacts. The next available location to begin storing the first contact within AA is at N + 1 . The new pointer value for the first contact detected will be set to NA(i) = N + 1 .

58. Therefore, after each cycle of iteration the value of the pointer NA(i) is the location within the array AA where the contact list information about element i begins. If the element i has no contacts, the value AA(NA(i)) will be zero, signifying the end of the contact list segment for element i . If element i does possess contacts, the value AA(NA(i)) will be set to $j + x_c/10000$. To find the end of the contact list segment for element i , it is necessary to skip through array AA by increments of five (starting at AA(NA(i)) until a zero is found. The next available location in AA will possess information about element i+1.

Example of contact list information

59. Consider the simple example shown in Figure 7a. At the start of the problem, two disc elements (2 and 3) are at rest on bar

element 1. An additional disc element (5) is resting upon bar element 4. Both bar elements are assumed to be fixed, i.e., they are prevented from rotation and translation. Assuming that the only forces acting on this system are due to gravity, the subsequent motions of the disc elements may be visualized according to the sequence indicated by Figure 7. Figure 8 shows the structure of the contact list for various stages of movement.



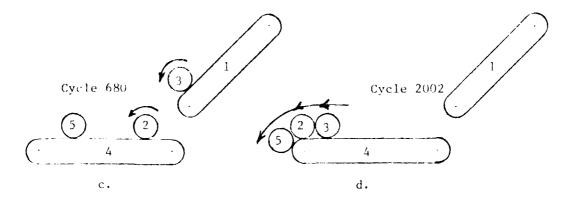


Figure 7. The contact list

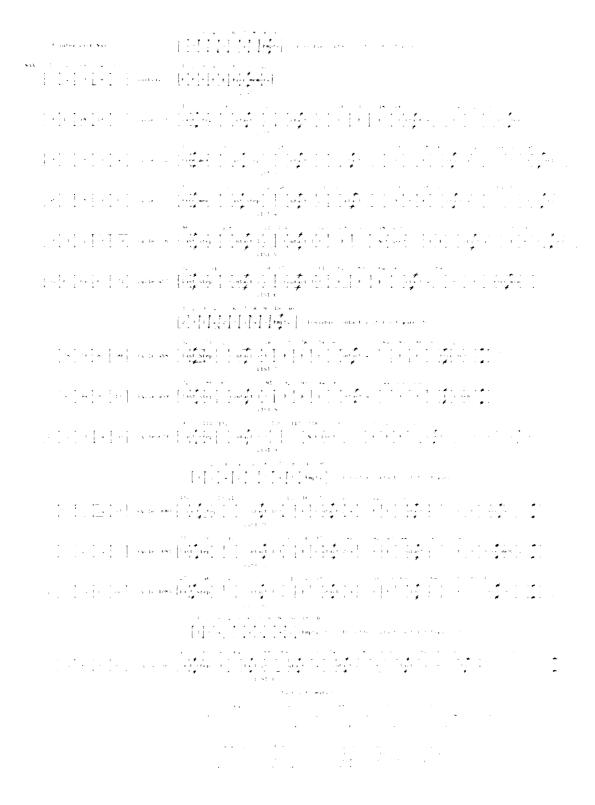


Figure 8. Organization of the contact lists

- proceeded by a call to subroutine PLIST. Subroutine PLIST creates the possible contact list it. The list LL as given at the top of Figure 8 is for the conditions shown in Figure 7a. The interpretation of the list is as rollows: elements 2 and 3 may contact element 1, element 2 has no (as yet undetected) contacts, element 3 has no (as yet undetected) contacts, element 5 may contact element 5 has no andetected contacts.
- 61. At the outset of any problem, the pointer list SA(1) is initialized to the value of -1 and the contact list is set to zero. Thus, initially, NA (element 1) = 1, NA(2) = 2,...,NA(5) = 5, and the contact data list (list 1, Figure 8) contains zeros as indicated by the symbol E (for empty). Since there are five elements in the problem, the first live storage locations are set to zero (as indicated by the five "E's" in list 1). The next available location for contact storage (at the next, or in this case the first cycle) is location 6.
- 62. The situation for the first cycle of MOTION is shown in list 2 (Figure 8). The pointer values NA are: NA(1) = 6, NA(2) = 17, NA(3) = 18, NA(4) = 19, and NA(5) = 25. Location 6 of list 2 is the beginning of the contact information for contact between elements 1 and 2. Immediately following the contact information for contact 1-2 is the information for contact 1-3 (starting at location 11 and ending at location 15). The zero (symbol E) embedded at location 16 signifies that there are no more contacts for element 1. Contact information for element 2 begins in the contact list at location 17 (NA(2) = 17). Location 17 contains a zero, which indicates that there are no contacts for element 2. Similarly, an contacts exist for element 3. The pointer for element 4 (NA(4) = 19) points to a nonzero value. Thus, the next five memory locations (19-24) yield contact information for contact 4-5. The pointer for element 5 (NA(I) = 25) points to a zero; therefore, there are no new contacts for element 5. Indeed, the last element may never possess any contact not already found.
- 63. A total of 20 memory locations were required to store all of the centacls for cycle 1. Since storage began at location 6, at present

- 25 locations have been used. Thus, the next available location for storage is 26.
- 64. At any given time cycle, it is necessary to retain the contact information about the previous time cycle. Only by retaining this information is it possible to check whether or not a detected contact was present at the last stage or if the contact is new. (If the contact is old, the normal and shear forces for the previous cycle are entered into the new contact list; otherwise, zero values for shear and normal forces are entered.) Figure 8 gives the new contact list for cycle 2 (list 3). List 3 is exactly the same as list 2 except the new contact list begins at location 26 and ends at location 45. It is readily apparent that all of these "new" contacts entered into the new contact list were present at the previous cycle. The pointer NA for a particular element is not updated to the new location until after it is used to check whether or not this same contact was present in the previous list. For example, contact 1-2 has been identified as a contact for cycle 2, and the contact information is to be entered at locations 26 through 30. The pointer value for element 1, NA(1), still has a value of 6. By going to location 6, it is noted that contact 1-2 was present in the previous contact list. The pointer value NA(1) is temporarily updated to a value of 11 so as to be directly pointing to the next location in the previous contact list where contact information about element 1 was stored. Thus, when contact 1-3 is examined (as to whether or not it was in the previous list), the pointer is in the proper position for quick recognition of this preexisting contact. After it has been determined that all contacts for element 1 have been exhausted, the pointer NA(1) is set to the new, current value of 26.
- 65. At each and every cycle an examination is completely carried out for every contact given in the <u>possible</u> contact list. information stored for the previous cycle is not used at all for determining whether

or not a contact is entered into the new contact list.* Only after it has been decided to enter a contact into the list is a scanning of the previous list required (and then only to decide what to do with the shear and normal forces). The method for updating the element pointer NA(1) simply provides an efficient method for checking on the presence of the contact in question.

66. The situation for cycle 3 is shown in list 4. Again, the organization is the same as for cycles I and 2 except the new contact list is removed by 20 more locations (i.e., locations 46 through 65). The situation for cycle 4 is not shown, but would be similar, except the new contact list would be stored in locations 66 through 85. This process could be carried out indefinitely; however, at some point memory would become exhausted. Since it is necessary to store contact information only for the previous and present cycles, a scheme was developed for recovering all available memory.

There is an option available that precludes the searching for contacts at each and every cycle. By entering a suitable variable (Command I in the Input Phase (Figure 9)), it is possible to stipulate that complete searches for contacts will be made only after every so many cycles. Unless otherwise stipulated through these options, a complete search is made for those element contacts in the possible contact list at each cycle. If the option is used, the program assumes that the current real contact list is to be used over and over; the shear and normal forces are being constantly updated. During this time, the contact coordinates and contact plane angles are properly computed. The use of this option can be dangerous since new contacts will not be detected until a complete search is called for. A complete search is always made upon each entry into MOTION. The use of this option can significantly reduce computer processing time since it negates the need for contact searches and for creating a new contact list.

67. Suppose that 120 storage locations* have been reserved for contact information as indicated in Figure 8. While processing in cycle 1, it is apparent that locations 1 through 25 are needed for the present and the previous contact lists. Thus, locations 26 through 120 are available as "free" memory for future cycles. While in cycle 2, the required memory locations are 6 through 45 (old plus new contact list). Therefore, the free memory consists of locations 46 through 120 plus 1 through 5. At cycle 3, the free memory consists of locations 66 through 120 plus 1 through 25. During cycle 4, the free memory extends from locations 86 through 120 plus 1 through 45. At cycle 5, the free memory consists of locations 106 through 120 plus 1 through 65. Thus, as cycle 6 is entered (and will require 20 locations to store the new contact list), it is apparent that the reserved memory space of 120 will

$$S = 5[(3xy - 2(x + y) + 1)] + xy$$
$$= 16xy - 10(x + y) + 5$$

Neglecting the last two terms, S equals 16 times the total number of elements. Technically, twice this amount of storage is required at any given cycle since the last contact list must be scanned to prepare the current contact list. In actuality, twice the amount is not needed because as soon as the new contacts are entered into the current contact list, there is no longer any need for the corresponding previous contact (as long as the sequence of insertion remains the same). At present, it is felt that a reserved space of $M \times 24$ should be adequate for most problems.

The size of common memory for the variables used in program DISC is set through the use of a Fortran PARAMETER statement. As seen from an examination of the program listing, the space allocated for the contact list AA is set to M x 24, where M is the maximum number of elements permitted for a problem. In practice, M will often be greater than the actual number of elements being considered, thus extra memory space is generally allocated. The common dimensions of M x 24 should be adequate for most problems. For example, consider a collection of closely packed uniform disc elements, x discs wide by y discs high. The total number of elements is xy. The maximum total number of contacts for this collection may be shown to be 3xy + 2(x + y) + 1. Since five storage spaces are required for each current contact (plus one additional space for each contact to store the zero or "E" marks) the storage S required to contain the current contact list is

be exceeded unless the front part of memory is reused. This condition can be avoided by computing the amount of free memory still available up to location 120 after each entry into the contact list. Since the next entry could require five additional storage locations, the beginning of the next entry is directed to location 1 if sufficient memory does not exist. Therefore, as shown for cycle 6, the first detected contact 1-2 is placed in locations 106 through 110, and the next contact placed in locations 111 through 115. The five storage locations available before location 120 are sufficient for storage of one more contact. However, contact 1-3 was the last contact for element 1. Therefore, a zero ("E") is placed at location 116, signifying the end of contacts for element 1. There are now only four available locations before the end of the reserved memory and, therefore, not enough locations to store information for a detected contact. The program now directs that storage begin at location 1. As shown for cycle 6, a zero ("E") is placed in location 1 (signifying that element 2 has no, as yet, undetected contacts). The pointer NA(2) is set to the value of 1. Locations 117 through 120 are not used.

68. As long as the contacts do not change, the calculations will proceed as described in cycles 1 through 6, each cycle requiring an additional 20 new storage locations of free memory. Cycle 457 is the situation just before element 2 rolls off bar element 1. Figure 8 presents the possible contact list existing for the conditions shown in Figure 7b. The interpretation of this list is as follows: elements ? and 3 may contact element 1, element 4 may contact element 2, element 3 has no undetected contacts, element 5 may contact element 4, and element 5 has no new contacts. The possible contacts 1-2 and 2-4 are included in the possible contact list; however, they will be excluded as real contacts in MOTION since the situation shown in Figure 7b is for the condition where element 2 is just breaking contact with element 1 and before contact with element 4. Figure 8 also shows the new contact list for cycle 458. Note that there are now only two contacts (contacts 1-3 and 4-5). The contact list for cycle 459 is just as it was for cycle 458 except the new contact is stored an additional 15 locations further

along in memory. Cycle 679 is the situation just before element 2 hits element 4. In addition, Figure 8 presents the possible contact list and the resulting new contact lists (cycles 680, 681, and 1085). Notice that again there are three contacts (contacts 1-3, 2-4, and 4-5).

69. In Figure 8, the <u>possible</u> contact list and the contact data list are given for some much later time (say cycle 2002). These lists are for the conditions depicted in Figure 7d. By this time, all of the disc elements are in contact and proceeding to roll off the end of element 4. At this stage, five contacts are detected (contacts 2-3, 2-4, 2-5, 3-4, and 4-5). Thus, 30 additional storage locations are required to store the contacts at each cycle.

Calculations of forces on the elements

70. As each element is, in turn, examined for contacts, the sum of all forces acting on the element is computed. Preceding each time cycle of iteration the force sum is set equal to the applied plus gravity loads, i.e.,

 F_{ysum}^{i} = vertical applied load minus the gravity load F_{xsum}^{i} = horizontal applied load F_{sum}^{i} = applied moment

In the event that the element being examined possesses no contacts, the computation proceeds directly to the calculation of accelerations and displacements (see Equations 6 through 9, Part II). If contacts are found for the element being considered, then the force sums are modified in accordance with Equations 1 through 5. The force sums are updated for both elements in contact. For example, suppose a situation exists in which element 3 makes contact with elements 5 and 7. When element 3 is examined for its contacts, the force sums will be updated twice (once for element 5 and again for element 7). During this process, the force sums for elements 5 and 7 are also updated. Then when element 5 is examined and it is determined that it has no, as yet, undetected contacts, the computation will proceed directly to Equation 6 since the force sum for element 5 was updated at the time when element 3 was examined. The same holds for element 7.

71. The computations of Equations 8 and 9 are particularly important. Equation 8 yields the new incremental element centroid displacements, and Equation 9 yields the new coordinates of the element centroid. It is these quantities that are required by Equation 1 for the next iteration cycle.

Other Subroutines

Subroutine PLTFOR

- 72. Subroutine PLTFOR is used to output the forces acting at the contacts between elements. Recalling that the contact list is structured such that each element has at least one entry (if no contacts were detected) and further that the storage is in sequential order, it is possible to, at any stage, thread through the contact list to obtain the contact forces. For each element i that possesses contacts, the contact list contains:
 - a. The number j of the element being contacted by i.
 - b. The 1D number for the contact (two bar elements in contact may have up to four contact points).
 - \underline{c} . The contact point coordinates, $x_{\underline{c}}$ and $y_{\underline{c}}$.
 - d. The normal and shear forces acting at the contact, F_n^c and F_s^c (stored as AA(K + 2) and AA(K + 3) in memory).
 - e. The angle to the contact plane, B.
- 73. These forces are output graphically. A pass is first made through the contact list to determine the largest absolute value of the shear or normal force in the list. This quantity is used to calculate a scale factor, and the vector representing this force will be 20 screen units long. The other vectors drawn to represent the shear and normal forces will be proportional. Thus, after each call to PLTFOR, a graphical representation of the contact forces are drawn on the screen. The contact coordinates \mathbf{x}_c and \mathbf{y}_c are used to locate the position of the vectors to be drawn, the shear force is drawn at the angle B, and the normal force perpendicular to the shear force. The user also has the option of obtaining a printout of the contact forces (Figure 9 or 10).

Subroutine LOOP

74. Subroutine LOOP is used to check whether or not the cross-hair cursor is located in the vicinity of an element centroid. In some cases, this test is used to modify certain commands.

Subroutine INTERVAL

75. Subroutine INTERVAL is used to change the frequency of returning control to the terminal. The normal method is to perform 250 time steps of calculations before halting calculations. At each halt the user directs what action is to be taken next.

Subroutine GRID

- $\,$ 76. Subroutine GRID causes a grid to be drawn on the screen. Subroutine SAVE
- 77. Subroutine SAVE may be called at any time by OUTPUT to store the current element geometry in memory. Subroutine SAVE is automatically called on each exit from INPUT. This feature provides the user a means of reworking the same geometry with, perhaps, a different set of parameters. Subroutine CIRCLE
- 78. Subroutine CIRCLE aids in the graphical output of disc and bar elements.

Subroutine SCREEN

79. Subroutine SCREEN is used to position the alpha cursor on the screen following input queries and some output functions. It prevents overwriting certain areas of the screen.

Subroutine GEN

80. This subroutine may be called during the INPUT phase. Its function is to create a collection of disc elements touching one another in a dense packing.

Subroutine VECTOR

81. This subroutine is used to vary the form of the graphical output. The normal mode of display is to draw the elements on the screen at their current positions. Subroutine VECTOR causes the display to track the element centroids.

Subroutine WEIGHT

82. Subroutine WEIGHT may be used to modify the weight of any or all elements.

Subroutine DAMP

83. Subroutine DAMP is used to change the damping constant K .

PART IV: PROGRAM OPERATION.

- to keep the data preparation simple and minimal. As a result, use of computer graphics was employed to a large extent. Most of the commands necessary to operate DISC consist of single keystrokes at the Tektroniz terminal. Figures 9 and 10 are the computer user's guides for the operation of DISC. Figure 9 gives a more detailed discussion of the user commands; Figure 10 is an abbreviated command listing. When one first uses the program DISC, the information given in Figure 9 should guide the user in the operation of the program. After the user becomes familiar with the program, the abbreviated command list (Figure 10) should suffice.
- 85. According to the user's guides, all input is initiated by entering a single character, following the appearance of the crossbair cursor. The crossbair cursor consists of two perpendicular lines crossing the screen of the terminal. The position of the intersections of the two lines is controlled by a set of thumbwheels on the terminal. In many instances, the entry of the single character is sufficient to accomplish whatever is desired. In other instances, certain parameters (or numbers) associated with the single character command are required. If the command requires one or more parameters, the alpha cursor (a small matrix of dots) will appear on the screen. As soon as the alpha cursor appears, the input parameters may be typed.
- 86. The detailed user's guide (Figure 9) is complete regarding the meaning of the various commands and, as such, will not be discussed further. Three examples will, instead, be described in order to illustrate the program.
- 87. Figure II is a series of computer drawings that describe a very simple problem. As soon as program DISC is called into execution, the message "INPUT PHASE" is written on the screen (Figure 11a). The collection of three elements was created in the following order:

100	********	INSTRUCTIONS FOR OPERATION OF PROGRAM "DISC" ***********
1961	6 16 1	Commands for Input Phase
104	· * * ·	after the appearance of the crosshai
201	4 44 1	Peremeters, is required, are entered atter appearance of the alpha . A cursor which will appear on the right side of the screen.
1000	* KEYParameters	ars Description
1444	***	Creates disc element centered at crosshair x cursor nosition
115 115 116	 ***	1st end of a bar element (semi-circle * centered at crosshair)
117 118 119 120	N	and end of a bar element. Note: repeated a hitting of the 2 key will create bar elements touching the the 2nd end of the preceeding a
122 122 123 124 124	~	bar element. * Creates a disc element adjacent to a preceding disc element in direction tadicated by the creates.
128	m m(3)	£
1111 133 130 130 130 130	* * * *	parameters. The lot element is centered at X the crosshairs. Parameters A of discs in bottom row
133 134 135	***	2nd # of rows 3rd increment for more or less discs(+or-)* for each successive row upward

Figure 9. Detailed instructions (Sheet 1 of 6)

136 *		**
137 *	Q	Draws all elements. Soreen will first erase X
138 ×		
140	U	Prints Coordinates of Indicated element *
145	αx	Changes value for Radius of discs and x
143 *		width of bar elements. Default value=20. x
145 *		screen units.
146 x 147 x	J(2)	Mocos the J-th sleasnt to new position x
¥ *		Redisplay by hitting D key
150 ×	P(2)	Re-Position all elements to new position x
151 × 152 ×		(given by parameters) relation to mignant * indicated to alement *
153 *		*
154 *	+ or;	Draws a Grid on the screen.
155 * 156 *	ш	Ends input phase. Calls compute phase. Also *
157 ×		causes a copy of the current geometry to be *
158 ×		stored in memory
160 x	I(1)	Sats interval between complete searches to *
161 *		detect contacts. Default value-1, maximum *
162 ×		** OS = en la c
164 ×	[backspace]	Recalls & displays the last stored copy *
165 X		

168 ×	Communication	Commends for Compute Phase
169 × 170 ×	1 2 4 4 1 7 2 4 4 1	# # # # # # # # # # # # # # # # # # #
171 ×	The following commands	The following commands are effective regardless of the position x
173 ×	or the Crossnair	K Curioor.

+7

Figure 9. (Sheet 2 of 6)

174 ×	Keyparameters	Description
		60- Calculate motions. Program calculates x the # of cycles given under I command (belowx Default interval is 250 cycles
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	I(1)	Sets Interval between program interrupts. * Program calculates the given # of cycles * before inquiring as to what action is to be * taken next. Default value is 250 curles
185 * 186 *	z	erases, all memory, to input phase.
* * * * * * * * * * * * * * * * * * *	vs	Saves current geometry and fixities. (Replaces the stored copy which was automatically saved upon exit from input x phase.
1000 1000 1400 1444	Q(1)	Damping factor-Parameter is X of critical X damping. Default value is 1.0
190	2	Zaroas all element velocities.
× × × 5	v	Draws a clock on the screen giving time from start of problem
* * * • • • • • • • • • • • • • • • • •	•••	Fixes all elements from movement
00000 0000 0000 04000 04×××	¥	for same
8 5 6 8	Q	Draws all elements
2116 2116 * *	O(letter)[where]	x Causes vector plot of contact forces to be x

Figure 9. (Sheet 3 of 6)

*******	* * *	• * * * * * * * * * * * * * * * * * * *	****	* * *	* * *	* *	* *	* *	* *
displayed. If printed output is desired, x position the crosshair to the upper right x side of Ewhere] printing is to start. Then hit the H key. Hitting any other key x will give plot only Causes output to be in vector mode, i.e. x instead of circles & bars being drawn, linesx connectig successive element centroid alternates modes back & forth.	or ; Draws a Grid	retur lling . Use ittin	e ONLY when the crosshall nits) on the centroid of ired.	Kegparameters Description	Fix element (from moving) in all directions X-Y-& angular.	Fix element from movement in X-direction.	Fix element in Y-direction	Fix element in angular (moment) direction	T U Unfix the element. Note: if any element has
	410								
00000000000000000000000000000000000000	000	0 0 0 0 0 0 0 0		33.0	444	9,0	24.4	4 4	40

19

Figure 9. (Sheet 4 of 6)

[space]	Erase (destroy) the indicated element from the system. Elements are re-numbered.
L(3)	Apply a load to a element at it's centroid ist parameter-X load 2nd parameter-Y load 3rd parameter-Moment Note: Each unit of load or moment is equivalent to the weight or moment of inertia of the standard 20 screen unit radius circle
U(3)	Sets INITIAL velocity in each direction. In screen units per second ist parameter-X velocity and parameter-Y velocity 3rd parameter-Rotational velocity
(B)	Sats velocity to prescribed CONSTANT value Parameters same as above. Prints coordinates of indicated element.
? or /[where]	ses centroid infor screen. After hitt sshair [where] out tput is of the for noX force vel

+ ×	z x x x x x x x x x x x x x x x x x x x	************	
* *	uingtons aro used oxantiy.		324
*	for either hit, the element centroid coor-		¥ ¥ טר טר
*	If crosshair is on/near an element centroid		321 ×
£ *	and and to (degrees) between them is original.		
*	hit A (or and key) at 2nd point. Distance		•
ł ×	two points. Hit > ker at ist point, then		
F×	Cond. to develope a constant of the second	<	317 *
×			316 ×
ļ	Default value is 1.0.		315 *
×p	elements weightless. The mass is not change		314 X
×	increase or decrease weight or to make all *		
*	Weight factor for element(s). Used to	6 (1)	
×		:	3112
*	values applies as for 0.		
×	Default value is 0.0. Same rule for negatives		
*	Tension value (force units) between elementsx	7(1)	
×			
*	ייבר ממטיין איז שפון אני דמ נשטיים איז מייבר ממטיין איז איז איז מיים על איז איז איז על איז		
۲,	Note of the coordination of the contract of th		
- 1	TOT BEARING		400
*	רמונה סו לאם שסטל אפקמלונים מאקום.		4 ÷
×	continuent is computed as the absolute		
×	Negative angles can be entered. If so, the		* * * * * * * * * * * * * * * * * * *
*	associated with the smallest angle is used) (C
×	different friction angles, the coefficient		
	(u*tan &). If two contacting elements have		× 262
*	Angle of friction in degrees for element.	0(zero)(1)	¥ 962
4 ×			295 *
← }			
*			• × ນ ຕ ກ ດ
92	NOT positioned on ANY centroid, the command applies to ALL elements	NOT positioned on	
*	ONLY that element is affected by the command. If the crosshair is	ONLY that element	
. ≯	STATE OF THE STATE	If the crosshair c	× 683
*	ands may be interpreted in two different ways.	The following comme	

Figure 9. (Theet 6 of 6)

+ |

	***************************************	``````````````````````````````````````
101 102 *	ABBREUIATED COMMAND	INSTRUCTIONS FOR "DISC"
103 * *	7	** ** ** ** ** ** ** ** ** ** ** ** **
105 ×		
106 ×	t phas	
	6 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
* ×	Koulparasetora	Description
		** **
* * *		Creates disc element
7 7 7 7 7	+ +	1st and of bar element x
115 x	a	- X Rond of Day alesent X
117 ×		
* * 00 0	Œ.	Places disc element Adjacent to preceeding X -
120 ×	B(3)	Creates Bunch of disc elements
-		
# ₩ ₩ ₩	Δ	Drøse pl 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
141 41	ů.	Prints coordinates of element
1 41 5 0 0 1 0 0 1	α	Change Radius of disc (width of bar).
1287 1287 * *	J[8)	Moves J-th elegat
1100 000 * *	(8)	- I BIBBBACO BILL BIBBBACO
4 50 44 44 44 44 44 44 44 44 44 44 44 44 44	ושו	Engle induct bracks. Creates a stores cover *

Figure 10. Abbreviated instructions (Sheet 1 of 3)

*	I(1)	Sats interval between contact searches.
* * *	[backspace]	•
4444 6444 6400 8 * * * * * * * * * * * * * * * * * * *		Compute The computed the compute
* * *	Key-paramotoro	Description
**:		Gocalculate motions
2011 2011 2011 2011	I(1)	Set Interval between queries (#-cycles)
Z * * 1		New probles. Restart from sorstoh.
· γ γ γ γ		Save (store) a copy of present geometry
155 × 0	0(1)	Damping factor.
× × :		Set all velocities to Zero.
× × +		Draws clock on screen.
* * * :		Tixos pil aleants.
× × :		Erase (Kill) soreen.
166 ×		Draws all alegents.
O * * * *	(latter)Ewherel	Output contact forces.
* * * *	0	Output to vector mode and vice verse
+ + × :	0 10	Grid 16 drawn.

Figure 10. (Sheet 2 of 3)

F X > E :	Fix element in all directions.
*****	Fix element in all directions.

****	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
****	rix in X-direction.
***	•
* * * !	Fix in Y-direction.
* * !	
.	Fix angular rotation.
M 4X	
185 × 1	
186 x [space]	Erase (destroy) element
×	•
188 * L(3)	Enter LoadsX-Y-Moment.
× ·	
190 x U(3)	Enter initial velocitiesX-Y-Angular
× :	
196 × c(3)	Set constant prescribed velocities
יינים או	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
104 × C	Print element centroid coordinates.
×	
196 x ? or /[where]	Print centroid information.
×	
198 x 0 (zero)(1)	Enter Friction angle- 0
S **	
+ +	
202 x U(1)	Enter Weight factor.
₩ ₩	· 1
204 # ^^	Print Distance between points.
205 ×	

Figure 10. (Sheet 3 of 3)

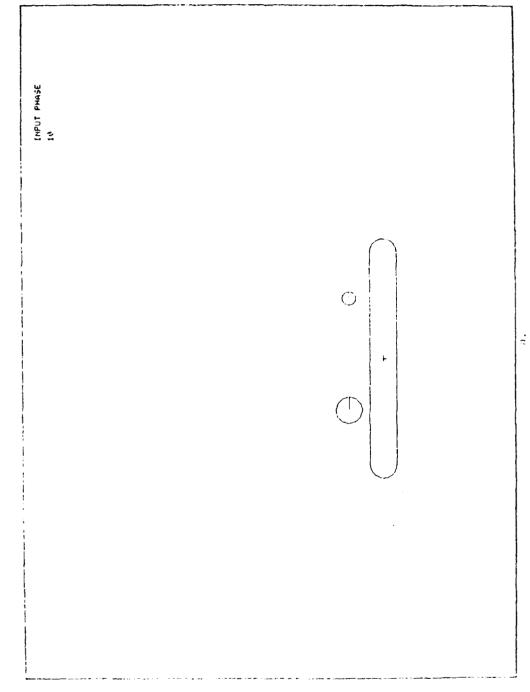


Figure 11. Example No. 1 (Sheet 1 of 10)

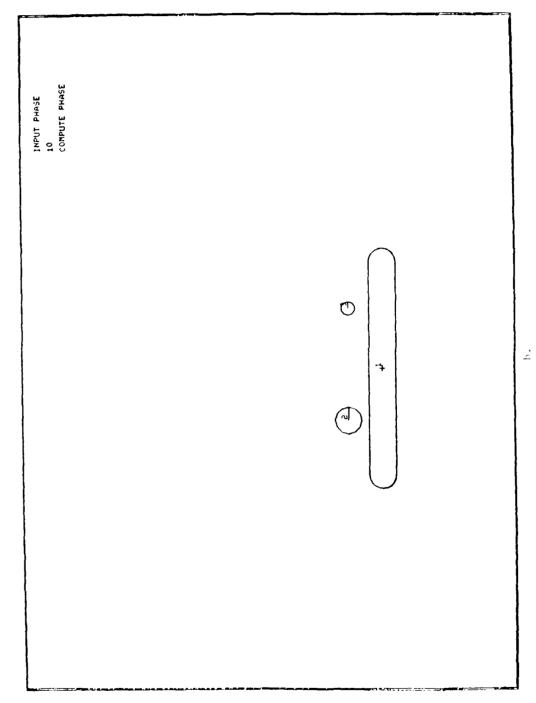
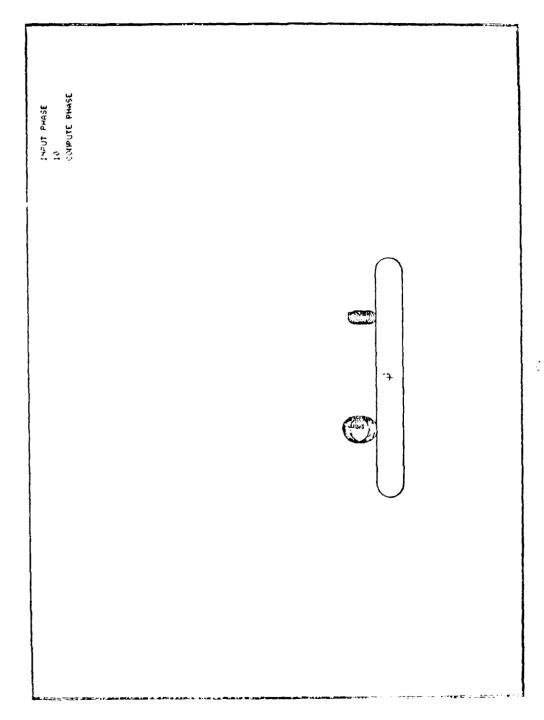


Figure 11. (Sieet 2 of 10)



Staroll, (Speet 3 of 10)

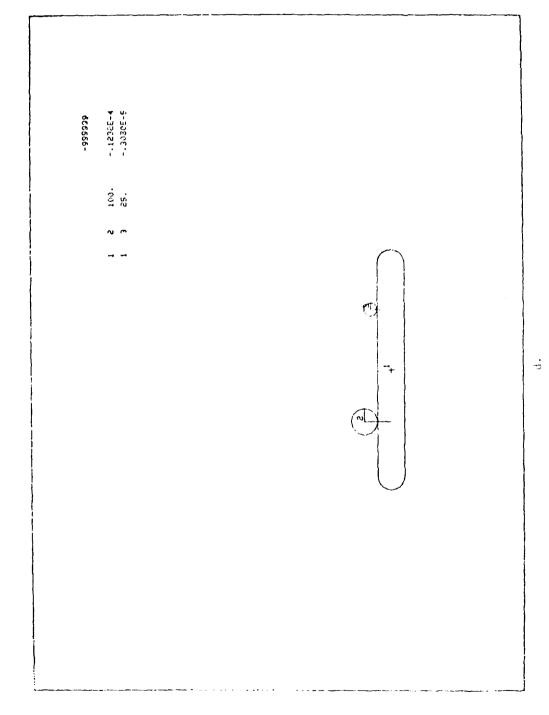


Figure 11. (Sheet 4 of 10)

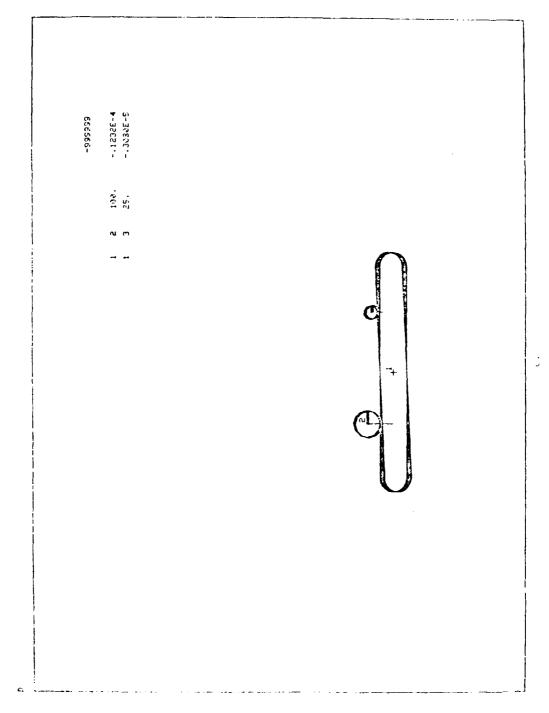


Figure 11. (Sheet 5 or 10)

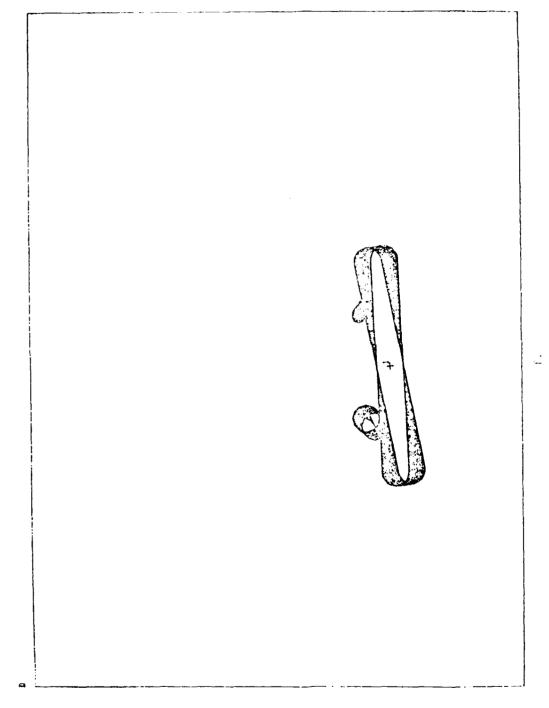
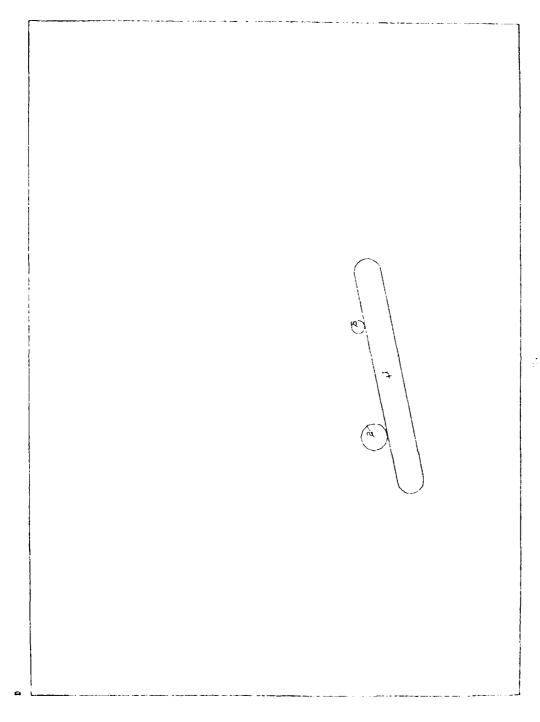
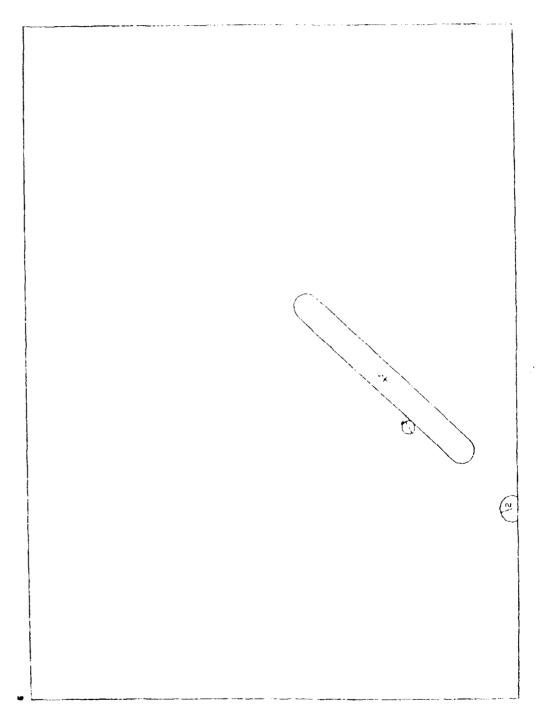
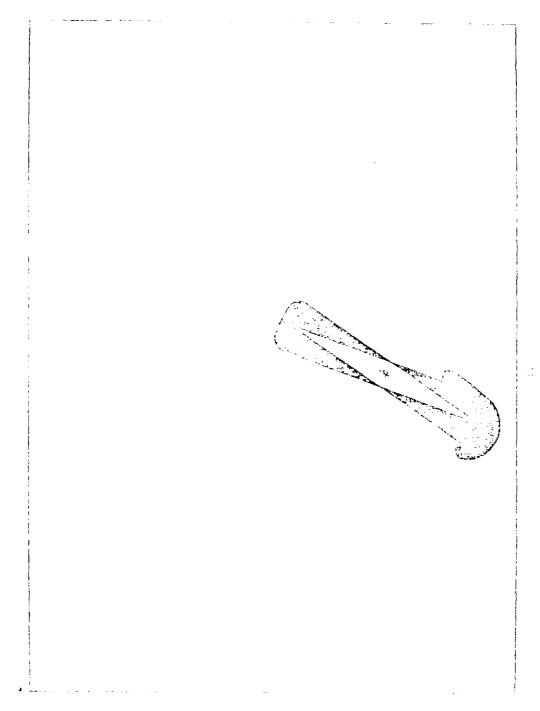


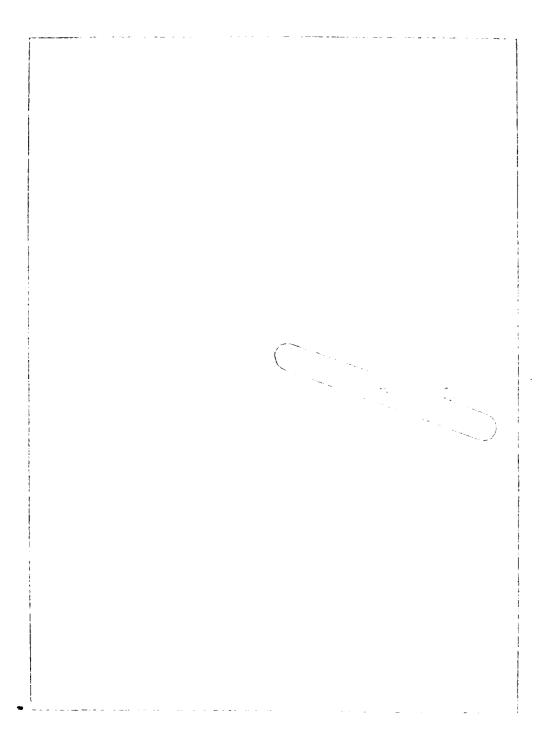
Figure 11. (Sheet 6 of 10)



Lare H. Back Car Section







- a. The bar element was created to positioning the crosshair cursor at the left end of decire) boottom of that element, and the ber "l" ... russ. The crossnair cursor reappeared. With the thomowhood, the crosshair cursor was moved to the right end of the per element, and the key "2" was struck. The bar element was then thown.
- b. The larger disc element was then are uted by postitioning the crossbair cornor at leaves, hed by tion, and the "spacebar" was struck. A disc with a rapper of 29 horeon units (the default value) was then draw at the leaf longshows in Figure 114.
- The smaller disc element was created by first billing the """ key and entacking the pareacter "10." unisorignals the computer to create any further elements with a radiation of so sever units (see appear finit corner of Figure 1991). The error materials of some placed of the location of the center of the small like and the "sparbar" struct.
- the compute plans of the program is striking as "as" larged at the compute plans of the program is striking as "as" larged at by, the message "COMPULE dead" appears on the series Colgues Hill. Modern point, the command "a" was given in: the elements were tedraw, and numbered. Following this step, the terrelement colored to establish is attained from all movement is peculifornize the cross-datic constraint as center element. Fairly and striking the key "as a contenct element.
- of 250 time steps of rotion. As clements move, their coefficient are drawn on the screen, the Tektronia terminal basis "herers" screen. The industry of the screens in the screens in the screens in the screens in the screens and in the entire screen is caused to be consect. Figure the shows the laters above motions of the two directors that under gravity to the later of element.
- 90. After a satisfification of 2." or "best to expect the street position of the satisfication of the satisfication of the satisfication of the street of the street position of the st

this time on, it is not possible to cause the small disc element 3 to separate from the bar element. The torces (in both normal and shear directions) are outputted to the upper right corner of the screen by the command "0" followed by "H." Thus, for the contact between elements 1 and 2, the normal force is 100.0 and the shear force very small (-0.1234×10^{-4}) . The normal force between elements 1 and 3 is 25.0, and the shear force is -0.308×10^{-5} . Recalling that the weight of the "standard" disc (radius = 20 screen units) is 100.0, the computed forces are correct. (Element 3 has a radius of 10 screen units and would weight only one-fourth the amount of the "standard" disc.)

- 91. At this point, the bar element is released from its fixity in all directions through the command "U" (with the crosshairs centered on the bar element). This command is followed by restraining the bar element from movement in both the horizontal and vertical directions by striking "X" followed by "Y" (again with the crosshair cursor on element 1). The bar element is now prevented from translating but is free to rotate about its center of gravity. To again begin calculations of motion, the "G" key is struck. Since the large disc 2 exerts a counterclockwise moment on the bar, the system begins to rotate as shown in Figure 11e.
- 92. Repeated "G" commands lead to the situation shown in Figure 111 and Figure 11g. The bar continues to rotate, and the disc elements roll down the plane. The lines joining the disc element centers to the periphery of the discs indicate the amount of rotation experienced by the discs (Figure 11g). At a later time (Figure 11h), the larger disc 2 has completely rolled out the bar element and will soon disappear from the bottom of the screen.
- 93. Recalling that, at an earlier time, it was specified that disc 3 should have a large mensile contact "strength," it should be expected that element 3 could not roll off the bar. Figure 11i indicates exactly that situation. Disc 3 rolls around to the underside of the bar element. Figure 11i is a refreshed plot of the final element positions occurring before calculations were terminated.

94. As a second simple example, consider the results given in Figure 12. In this example, three problems are being solved simultaneously. As soon as the input phase is entered, the 'ey "4" is struck and couses a grid to be drawn on the screen. The grid is helpful to orient the three long bar elements at a 45-deg angle (Figure 12a). After inputting the three bar elements and the two disess, the compute phase is called and the elements are numbered (Figure 12b). All of the lower bar elements are then fixed by typing "F" in sequence for those elements. Then, element 4 is restrained from rotating by placing the crosshair cursor at its center and striking "9." After typing "G," the upper elements fall and proceed to stide down (element 2 rolls down) the long bar elements (Figure 12c). The default value for coefficient of friction is 0.5; thus the shear forces should never exceed one half of the normal forces. Figure 12c gives the normal and shear forces at all contacts. In every instance, the shear force is one half of the normal force. Notice that there are two 5-6 contacts, the first one being a larger value than the second. This is prelictable since most of the weight should be felt by the lower contact.

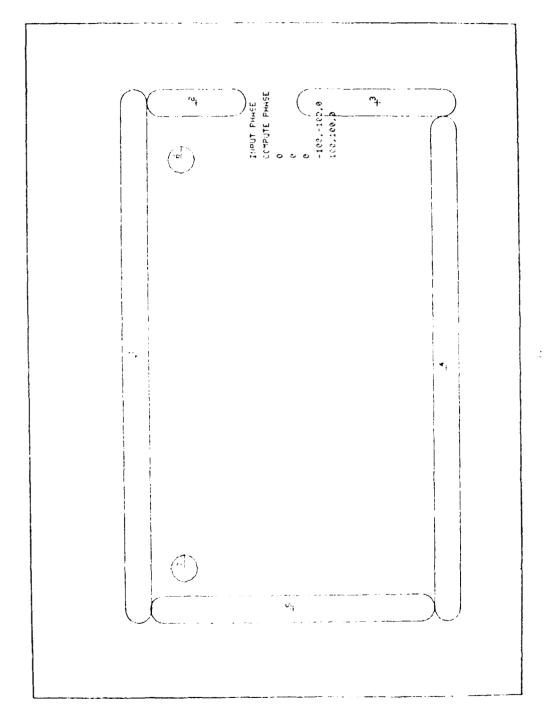
195. The third example is illustrated in Figure 13. The first tigure of this series shows a situation in which disc elements 6 and 7 are surrounded by a boundary of fixed for elements. There is, however, an opening between elements 2 and 3. When describing the locations of the elements in the input phase, the example phase is entered. Therents 1 through 5 are immediately fixed through the arc of the "F" key. Following that step (Figure 13a), the weights of all element care set to zero by the use of the "W" key (the masses reason unsharead). For, the damping constant is reduced to zero is using the "O" mequence. Next, the coefficient of triction is set to zero (asing the "\$" key) so that the angles of reflection will equal the angles of incidence in whatever motions ensue. Through the ase of the "V" key, the initial velocity of element 6 is set to $v_{\rm m} \approx 100$ screen unit tiper second, $v_{\rm m} \approx -100$, and the angular velocity is set to zero. For slowent 5, the "V" key was used to set $v_{\rm m} \approx 100$ units, per second and $v_{\rm m} = 100$ units. Figure 15b

, y

Figure 19. Pengole No. 2 (Spect Let 2)

Haure 12. (Sheet 2 of 3)

Figure 12. (Sheet 3 of 3)



Trend to the artists of Stock Lot 3)

A SHARE THE RESERVE OF THE SHARE THE

Transity, (Sloet 3 of 3)

shows the resulting motions after striking key "G." Notice that disc 6 proceeds downward to the left. Disc 7 starts upward to the right, strikes the upper boundary, and turns downward to the right. Following the first calculation cycle of 250 time steps, output is converted to the "vector" mode by hitting key "*." When in the vector mode, the elements are no longer drawn. The output now consists only of the movement of the element centers. Element 7 proceeds to rebound off the lower boundary, then strikes the lower end of element 2, and exits through the opening (Figure 13b). Element 6, however, does not quickly find an exit through the opening. In Figure 13c, element 6 suffers a total of 21 rebounds before hitting the rounded portion of element 2. Element 6 then is deflected in an almost horizontal direction to the left wall and shoots across and out through the opening (almost dead center).

96. The purpose of the three example problems is to demonstrate the flexibility and range of this program. Problems involving quasistatic behavior and large rapid motions are included within the same framework. Indeed, at one time, this program contained options for including "inverse square of the distance" forces so that the user could compute orbits of gravitational bodies. These options are not germane to the general fields of geotechnical engineering and, as such, are not present in this version. However, the program DISC is basically a tool that will solve many fundamental problems in rigid body mechanics.

PART V: CONCLUSIONS AND RECOMMENDATIONS

Conclusions

- 97. A computer program entitled DISC has been developed for analyzing disc (and bar) shaped elements. The formulations, based upon the distinct element method, allow for large, rapid movements, or quasistatic movements. The discretized behavior of the elements are kinematically calculated by the program DISC so that individual system particles (the discs) are capable of rolling, aliding, displacing each other, etc. The distinct element method provides a contrast to common continuum analysis procedures, such as the limite element method.
- 98. At the beginning of this study, the writer was optimistic that this method would be directly applicable to problems encountered in geotechnical engineering. During the course of this study, rany "conputer runs" were made in the hope of modeling a variety of cormon geotechnical phenomena, such as slope stability, retaining walls, learning capacity, and pile driving. Home or these attempts were satisfactor to the extent required for reporting in this document. That is, the writer has not been able to extend the method to common geotechnical (rob) as in a direct tashion. However, many geotechnical problems (a). To physically modeled by an assemblage of discs. The results of a new modeling coull lead to (at the present) an understanding of the letails of the individual particle motion. For example, a retaining wall ould be physically modeled by an assemblage of discs, as could a projective penetration problem. The program DISC is an extremely powerful model to analyze disc-shaped objects. To this purpose, the writer inews or no other method that is nearly as accurate or flexible. For example, a paper was recently published describing a program garte limitar to bisc (Candall and strack, 1979). In this paper, the comparison was made between a computer modeling program entitled BALL (similar to 1980) and photoelastic studies of physical dises. The comparison was excellent; however, once again, both studies were sodeling class.

method to more meaningful situations. It is the writer's opinion that this type of analysis will eventually be adopted in many areas of engineering. The concept of analyzing discrete particles in an explicit scheme is attractive to the formulations of geotechnical problems.

Recommendations

written to be quite general. It a subsequent user of this program has to analyze a particular problem, he should make an attempt to modify the basic, general program to his needs. For example, if the problem requires only a quasi-static analysis, many savings could be accomplished by fewer searches for near contacts. Researchers generally tend to direct new concepts to a large audience and always try to account for any possible application of their developments. A user of this research would be well advised to simply store a copy of the original program DISC within his files and then modify and enhance whichever portions of DISC that fit his applications. That is, adapt the program to the particular need.

REFERENCES

Condall, P. A. 1971. "A Computer Model for Simulating Progressive Large Scale Movements in Blocky Rock Systems," Paper 11-8, Proceedings of Symposium on Rock Fractive, International Society for Bock Mechanics, Namey, France.

Cundall, P. A. 1974. "A Computer Model for Rock Mass Behavior Using interactive Graphics." Technical Report 2-74, F. S. Army Corps of Engineers, Missouri River Division, Omaha, Nobr.

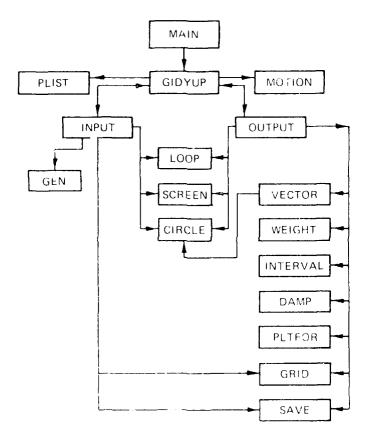
Cundall, P. A., and Strack, O. a. L. 1979. "A Discrete Numerical Model for Granular Assemblies," Costechnique, V-1-29, No. 1, pp 47-65.

Vocatele, M. D. 1979. "Rational Design of Tuanel Supports: An Interactive Graphics Based Analysis of the Support Requirements of Excivations in Wointed Rock Masses," Toological Report G1-79-15, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss.

West Point Military Academy and U. S. Army Engineer Waterways Experiment Station. 1975 (revised 1979). "Primer or Computer Graphics Programming," U. S. Army Engineer Waterways Experiment Station, Ch. Vicksburg, Miss.

AMPENDIA II: SCHEMATIC DINGRAY OF DISC AND FORTRAN SOURCE LICTURE

SCHEMATIC FLOW OF DISC



```
U(I)=(R(I)=R(I)===1415926+R(I)=5(I)=2.)
1. /12.5663706
25 xm(I)=u(I)-AGRAU
        VERVI
150 CALL CIRCLE(X(N), 4(N), NR, S(N), T(N))
N=N+1
                                                    10 TO 26
```

```
ハメのあつ "それをと "メのだメ"といると、いいのの、いいのに
THE TANK THE
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   F.
                                                                                                                                                                                                                                                                                                                                                                                                                                                          | March | Marc
```

```
L UMOVE(XCMU),VCMU))
**-57.295795#AMOD(TCMU)—TIN(MU),6.28318531)
L UROTAT.#301
MXX(XC)-6504.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ### X(1) *X(1) *X(
                       THE CALL OF THE CALLEGE CALL OF THE CALLEGE CALL OF THE CALLEGE CALL OF THE CALLEGE CA
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                146 XEBX*XEIV*10000.
166 YEBIX*YCHUO000.
166 YEBIX*YCHUVUO000.
166 YEBIX*YCHUVUO000.
168 HE CONTINUE
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           CALL UPRINTCK(1)+3., V(1)+3.,XI:
4 CONTINUE
4 TO TO 180
11 CONTINUE
           RESTRICTOR (2013) GO TO 35
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             COSTITUTE TO BE LEVEL TO SEE THE SEE T
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 EG. 60 40 5
```

At.

```
6 C-FLCAT(J)/R.0

6 O-FLCAKB-C)*RRSIGK(1.0,D(B))-(R.0RBERD(B))

6 100 Y-SORT(B.0)*RR+V;

7 DO B00 I-ST,N-1

100 CALL UCRCLE(X(I),Y(I),R)

CALL USET(TEXT-)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 10 SUBROUTINE WEIGHT(EF)

20 CALL SCREEN(XP, VP)

40 CALL WEAD(XP, VF)

10 CALL WEAD(XP, VF, VF)

10 RETURN
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          30 SUBROUTINE INTERVAL(KASK)

10 CALL SCREEN(XP, VP)

10 CALL USET("REAL")

10 CALL UREAD(XP, YP, X, 1., FL)

10 RASK X4+1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                           SUBROUTINE DAMP(DF)
CALL SCREEN(XP,YP)
CALL USET(*REAL*)
RETURN RETURN
RETURN
### DO 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
```

```
| Mark | 
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        60 CONTINUE
6 766 CONTINUE
8 ICTOCCE.EQ.1) GO TO 765
8 NILENDO
8 NILENDO
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         ÎFÎICYCLE.ME.13 GO TO 700
NUM-0
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    MET CAN
```

 ΔJ

```
9999 YC-XXEGO-YNESI+XI

9710 | Frite Coils | Frite Frite Coils | Frite Frite | Frite F
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             GT.N. GO TO 188
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        CONTINUE
```

MRET .:

17.2. .NE.IX) 3-8-PI

K) B-B-PI

#XI+R(IK)#SIN(W)

```
10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 | 10050 
 200 CONTINCE
```

```
### 10 CONTINUED TO THE PROPERTY OF THE PROPER
```

11.7

In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Inducertics, John B.
Amazyni, Then, appeard laste of Highr engly which intermetive simplifiers by John B. Calmenters, The desirable of an interpretation of the most my John B. Calmenters, Experiment State in a processor in a April State in a processor in the State in a April State in

77, Tolp. of 181. ; John -- (Misser Pane of per 182). Army Empineer Waterway: Experient Station ; S-1 --Symptitle.

"Lesember 1989."

Fefferencest p. 77.

reformments p. F.

1. Impatien consister. C. Consister consenses. Mill.

1. mited Mtanus, representation for the long formation with wellinvestigated. The following Method. Applications with with
magnetizate Mtation. In technical one matches the following
IV. Conference of Miller parameters of miles of the constant
Waterways Experiment Stations () H = - c. o view (fig.) and a first of the first of t

!

